Utilisation et Description du Code d'évolution stellaire CESAM2K

Pierre Morel
OCA Observatoire de Nice, BP. 4229, 06304 Nice Cedex 4
UMR CNRS 6529

EN COURS D'ÉLABORATION

Version V2, 11 Avril 2006.

Quand un jour, tôt ou tard, il faut qu'on disparaisse,
Quand on a plus ou moins vécu, souffert, aimé
Il ne reste de soi que les enfants qu'on laisse
Et le champ de l'Effort où l'on aura semé.

Ch. De Gaulle

Table des matières

Table des figures

Liste des tableaux

Index

Première partie

IMPLANTATION et EXPLOITATION

Chapitre 1

La version 2000 de CESAM

On n'exécute pas tout ce qui se propose

Et le chemin est long du projet à la chose.

Le Tartuffe.

1.1 La restructuration

En vue d'extensions futures, une restructuration[1] de CESAM est apparue nécessaire. Elle est articulée autour d'une réécriture du code en `FORTRAN95` qui permet le regroupement des routines en `MODULE`s[2]. Elle concerne les points suivants :

- dépoussiérage et fiabilisation des algorithmes,
- suppression de variables et de routines obsolètes,
- amélioration de la lisibilité, en particulier par l'utilisation de tableaux à plusieurs dimensions qui, grâce aux `MODULEs`, sont gérés de façon plus fiable et surtout plus simple par `FORTRAN95` que par `FORTRAN77`,
- exploitation sous forme "générique" des routines de physique, afin d'éviter les arborescences d'`EXTERNAL`s et ainsi de faciliter et fiabiliser la mise en œuvre des extensions et des modifications ultérieures,
- utilisation de facilités de programmation offertes par `FORTRAN95` par exemple : allocation dynamique des tableaux, boucles infinies, opérations sur les tableaux, gestion des chaînes de caractères, etc...
- CESAM2k est un abrégé de CESAM2000.

On trouvera § 1.1 (Page 7) un organigramme général du code.

[1] Ce paragraphe concerne principalement les utilisateurs habitués aux versions précédentes de CESAM.

[2] Un module est une sorte de "super `COMMON`". Un des intérêts de cette structure réside dans le fait que le compilateur vérifie la concordance des types, (`real, integer...`), du nombre, des dimensions des arguments des listes d'appel des routines entre le programme appelant et la routine appelée.

1.2 CESAM2*k*

Le principe de l'exploitation de CESAM2*k* est différent de celui utilisé dans les versions précédentes de CESAM. L'exécutable, `cesam2k.out`, est créé une fois pour toutes. C'est dans le fichier de données, `mon_modele.don`, que la physique à utiliser est indiquée conjointement aux principaux paramètres définissant le modèle à calculer. Il n'y a donc plus à créer d'exécutable adapté à chaque utilisation, comme c'était le cas avec les versions précédentes de CESAM. Il est toutefois possible de modifier la plupart des options retenues dans la source, grâce à des fichiers de "réglage" décrits au § 4 (Page 37), qui permettent d'adapter CESAM aux moyens de calcul dont on dispose et, dans une large mesure, aux particularités des modèles à calculer.

CESAM2*k* est programmé en `FORTRAN95`, ce qui permet un dimensionnement dynamique des tableaux adapté à chaque cas particulier. Les compilateurs suivants ont été utilisés avec succès :

- sous LINUX : `ifc` de INTEL, Lahey-Fujitsu `lf95`,
- sous UNIX : compilateur UNIX des machines ALPHA
- sous WINDOWS : Compact Visual Fortran

La signification du numéro de version `CESAM2k_Vi.j.k` est la suivante :

i augmente d'une unité à chaque modification du contenu des fichiers binaires de reprise *.pms, *.rep, *.dat, *etc...* les indices suivants **j** et **k** étant remis à 0.

j augmente d'une unité à chaque modification du contenu du fichier de données *.don, n'impliquant pas sa remise à 0, l'indice suivant **k** étant remis à 0.

k augmente d'une unité à chaque modification notable n'impliquant pas sa remise à 0.

Dans le sous-directory `SOURCE`, le fichier "`journal`" contient une description succincte de la chronologie des aménagements effectués.

PB `Le compilateur ifc de INTEL`[3] `n'a intégré les recommandations 1998 de la norme, qu'après mars 2004. Des difficultés de compilation peuvent apparaître avec les versions antérieures,` *cf.* § A.1 (Page 221).

PB CESAM2*k* `utilise le logiciel de dessin PGPLOT dont le fichier de configuration n'est pas prévu pour le compilateur ifc. Thierry Corbard, Observatoire de la Côte d'Azur, a construit les procédures nécessaires que l'on trouvera dans le sous-directory SCRIPTS.`

1.3 Contenu de la distribution

Le package `CESAM2k`[4] contient les sous-directory :

- `EXPLOIT` : ensemble de programmes et d'exemples de fichiers pour l'exploitation.
- `NOTICE` : sources LaTeX et fichiers postcript de la notice d'utilisation et de l'aide-mémoire.

[3] Contrairement aux idées reçues, ce compilateur n'est pas gratuit, cf. son contrat de licence.
[4] 2k pour "2000".

- SCRIPTS : procédures UNIX/LINUX pour l'installation et l'exploitation avec le shell csh.
- SOURCE : ensemble des routines, modules et programmes de la source.
- SUN_STAR_DATA : données physiques, tables d'EOS par exemple.
- TESTS : programmes de test[5].

Dans le sous-directory SOURCE, un fichier nommé journal recense les divers aménagements effectués à partir du 01/05/05.

1.4 Installation

On décrit l'installation sous UNIX/LINUX en utilisant le C-shell.

PB Avec un compilateur dont la version n'est pas à jour de la norme F95, (comme certaines versions du compilateur d'INTEL) il y aura refus d'effectuer une allocation dans une routine autre que celle où est déclarée l'allocation. Il y a lieu de remplacer dans les routines concernées, les instructions ALLOCATABLE par POINTER pour les tableaux incriminés *cf.* § A.1 (Page 221).

- Détarer la distribution :

  ```
  tar -xzvf CESAM2k.tar.gz
  ```

  Le sous-directory CESAM2k est créé. Il contient 6 sous-directory tarés et compressés :

  ```
  SOURCE.tar.gz
  EXPLOIT.tar.gz
  SUN_STAR_DATA.tar.gz
  SCRIPTS.tar.gz
  TESTS.tar.gz
  NOTICE.tar.gz
  ```

  Il convient de les détarer au fur et à mesure des besoins. Cette disposition permet de placer ces sous-directory aux endroits les mieux adaptés et ainsi de faciliter les mises à jour qui, la plupart du temps, n'affectent que le seul sous-directory SOURCE.
- Dans le sous-directory SCRIPT, adapter au compilateur F95 utilisé, les paramètres des procédures de compilation compile2k et d'exécution exe2k ; y indiquer aussi les chemins de la source et des bibliothèques. Rendre les procédures exécutables : chmod 700 *.
- Pour une utilisation avec le shell csh ou tcsh, dans le fichier .cshrc ajouter les chemins de la bibliothèque et des exécutables, par exemple :

  ```
  set mypath=(.... ~/CESAM2k/SOURCE ~/CESAM2k/EXPLOIT .....)
  set path=(. ~ ~/SCRIPTS $path $mypath) .....).
  ```

  Avec le shell bash, dans le fichier .bashrc, ces instructions deviennent :

  ```
  export
  PATH="$PATH : /CESAM2k/SCRIPTS : /CESAM2k/EXPLOIT : /CESAM2k/SOURCE"
  ```

[5] Certains de ces programmes sont à adapter à la présente version de CESAM2*k*.

– Si ce n'est déjà fait, installer le logiciel de dessin `PGPLOT`[6], et indiquer le chemin de sa bibliothèque dans le fichier `.cshrc` :
  `LD_LIBRARY_PATH **** :/usr/lib/pgplot`
  et définir la variable d'environnement[7] :
  `setenv PGPLOT_DIR /usr/lib/pgplot`.
  On donne au § A.2.1 (Page 222) un supplément d'informations concernant l'installation de ce logiciel.
  A défaut, de ce logiciel de dessin, dans le sous-directory `SOURCE` :
  – Supprimer `boite`, `box`, `des_r`, `des_m` de la liste des `INCLUDE` du module `mod_numerique`.
  – Dans la subroutine `des`, ne conserver que `CASE ('no_des')` et, éventuellement, `CASE DEFAULT`.
  – Dans la subroutine `sortie`, supprimer `CALL pgend`.

PB `La liste de ces corrections n'est peut être pas complète.`

1.5 Adaptation de la source

Avant de créer le module exécutable, il convient d'adapter certains paramètres libres de la source à l'installation dont on dispose.

– Dans la routine `des_m`, ligne 105, adapter le nom du "device" : `device='/xw'`, éventuellement coder `device=' ?'` de façon à obtenir la liste des devices disponibles.
– Faire de même dans la routine `des_r`, ligne 89.

Dans les routines `des_m` et `des_r` il conviendra, par la suite, d'adapter la hauteur et la largeur des cadres, de façon à optimiser leur centrage dans la fenêtre dont on dispose, ou encore de personnaliser le dessin *cf.* § 4.10 (Page 47).

1.6 Création du module exécutable

Se placer dans le sous-directory `SOURCE`, y lancer la procédure[8] `genere_cesam2k`. Après la compilation, on obtient la liste des "`object module`" de la bibliothèque `libcesam2k.a` et le menu d'entrée de CESAM :

```
mod_kind.o
mod_numerique.o
mod_donnees.o
mod_variables.o
mod_etat.o
mod_opa.o
mod_conv.o
mod_atm.o
mod_nuc.o
```

[6] Avec son mode d'emploi, le logiciel `PGPLOT` est téléchargeable via `anonymous ftp` à l'adresse `http ://astro.caltech.edu/~tpj/pgplot/`.

[7] Une de ces instructions est peut être redondante. L'avis d'experts est accepté.

[8] Dans la suite le symbole <== indique une instruction à saisir.

```
 mod_bp_for_alecian.o
 mod_evol.o
 mod_static.o
 mod_cesam.o
 mod_exploit.o
 Encountered 0 errors, 0 warnings in file cesam2k-dbg.f.

CESAM speaks a bit of english if you include in
the working directory a file named langue with the statements :
&NL_LANGUE
langue='english'
/
cf. aide_mem2k, chapter Personnalisation

Pour arrêter : taper 0 puis RETURN
Pour poursuivre une évolution : taper 1 puis RETURN
Pour initialiser un modèle de ZAMS : taper 2 puis RETURN
Pour initialiser un modèle de PMS : taper 3 puis RETURN
 ARRET
```

Suivant l'installation et les paramètres de compilation retenus, la création de la bibliothèque et du module exécutable nécessitent de l'ordre de 3 à 10 minutes.
De façon similaire, on peut créer un exécutable de debug avec la procédure `genere_cesam2k-dbg` du sous-directory `SCRIPTS`.

1.6.1 Utilisation d'un MAKEFILE

Il est évidemment possible d'exploiter CESAM2k via un `MAKEFILE`. Th. Corbard a construit celui donné en exemple au § A.3 (Page 223). Pour l'utiliser, après avoir adapté les options de compilation, déplacer le fichier Makefile du sous-directoruy `EXPLOIT` le sous-directory `SOURCE`. Adapter les arguments au compilateur utilisé et les divers chemins. Enfin, saisir la commande `make`, automatiquement le module exécutable `cesam2k.out` sera créé.

1.6.2 Fonctions des modules

Les modules de CESAM2k ont les fonctionalités suivantes :
- Module `mod_kind` : regroupe les types des variables.
- Module `mod_numerique` : regroupe les routines purement numériques.
- Module `mod_donnees` : regroupe l'ensemble des quantités fixes au cours de l'évolution.
- Module `mod_variables` : regroupe l'ensemble des quantités variables au cours de l'évolution.
- Module `mod_etat` : regroupe les routines concernant l'équation d'état.
- Module `mod_opa` : regroupe les routines concernant le calcul de l'opacité.
- Module `mod_conv` : regroupe les routines concernant la convection.
- Module `mod_atm` : regroupe les routines de restitution de l'atmosphère.

- Module `mod_nuc` : regroupe les routines concernant les réactions thermonucléaires.
- Module `mod_bp_for_alecian` : regroupe les routines concernant le calcul des accélérations radiatives, suivant le formalisme de G.Alécian.
- Module `mod_evol` : regroupe les routines concernant l'évolution temporelle de la composition chimique.
- Module `mod_static` : regroupe les routines concernées par la résolution de l'équilibre quasi-statique.
- Module `mod_cesam` : regroupe les routines concernant la gestion du calcul.
- Module `mod_exploit` : regroupe des routines concernées par l'exploitation des résultats.

On trouvera § 1.1 (Page 7) un organigramme général du code. Chaque routine `PRIVATE` et/ou `PUBLIC` est introduite sous la forme d'`INCLUDE` dans son module d'appartenance, ce dernier étant indiqué dans les commentaires du début de la routine.

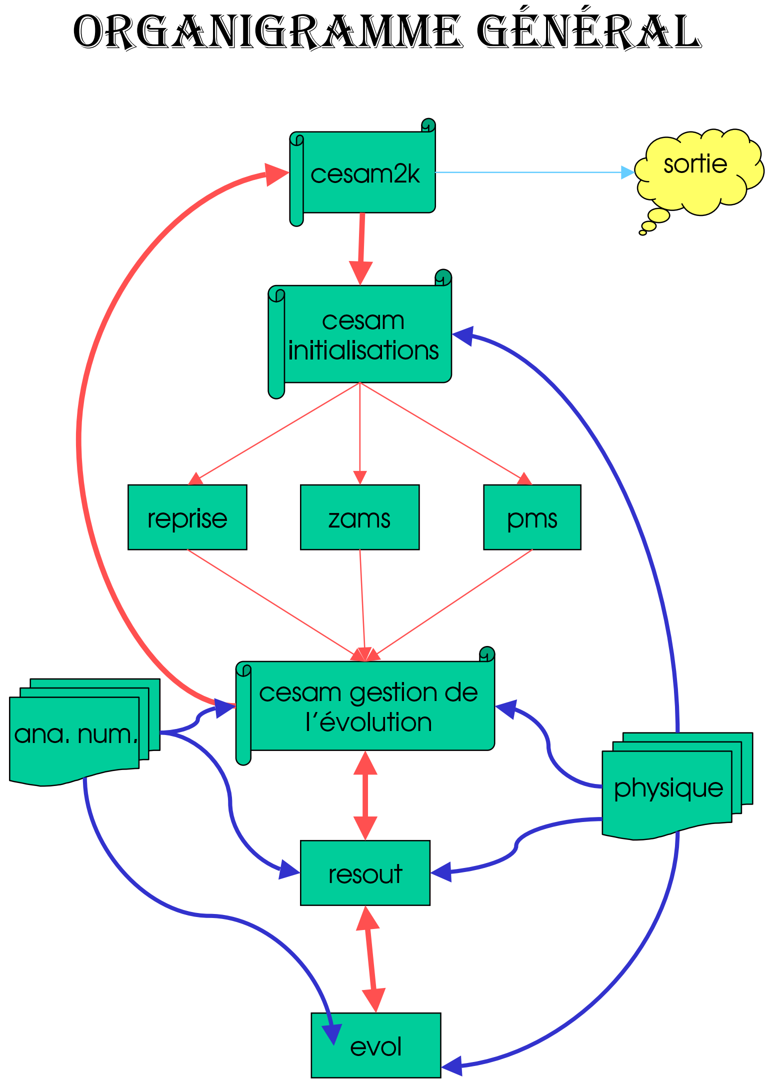

FIG. 1.1 – Schéma de l'organigramme général de CESAM2k. Le programme `cesam2k`, compilé une fois pour toutes, fait appel à la routine `cesam` qui constitue en fait, le programme principal. Dans `cesam`, après des initialisations, on sépare le traitement des 3 possibilités : poursuite d'une évolution, initialisation sur la ZAMS homogène, initialisation d'une PMS. Le calcul se poursuit par des opérations de gestion de l'évolution, listing, création de fichiers de sortie, dessin, etc... Est ensuite fait appel à `resout`, où les équations de l'équilibre quasi-statique sont résolues en alternance avec celles de l'évolution du moment cinétique et de la composition chimique. Ces dernières sont résolues séparément dans `evol`. Une fois la convergence obtenue, il y a retour vers les algorithmes de gestion de l'évolution de `cesam` et, de là, poursuite de l'évolution ou retour à `cesam` pour sortie. Les routines d'analyse numérique et de physique sont sollicitées à différents niveaux de calcul.

Chapitre 2

Implantation

2.1 Test de fonctionnement

Copier les fichiers `mon_modele.don` et `m010.zams` du sous-directory `EXPLOIT` dans le sous-directory `TESTS` dans lequel on se placera. Editer le fichier `mon_modele.don`.

- Remplacer `"des_m"` par `"no_des"` si `PGPLOT` n'est pas opérationnel.
- Adapter `"nom_chemin"` au chemin du sous-directory `SUN_STAR_DATA` de votre installation.
- Rétablir les instructions suivantes :

```
&NL_CESAM
NOM_CHEMIN='~/SUN_STAR_DATA/',    <==== à adapter
NOM_CTES='ctes_94',
NOM_DES='des_m',   <==== à adapter
NOM_OUTPUT='no_output',
N_MAX=2000,
PRECISION='np'
/
&NL_MASS
MTOT=1.d0,
NOM_PERTM='pertm_ext',
MDOT=0.d0
/
&NL_EVOL
AGEMAX=1.d2,
ARRET='else',
DTLIST=1.d10,
LOG_TEFF=10.d0,
NB_MAX_MODELES=200,
HE_CORE=-1.d0,
T_STOP=5.d7,
X_STOP=-0.1d0
/
&NL_CHIM
GRILLE_FIXE=.FALSE.,
```

```
NOM_ABON='solaire_gn',
MODIF_CHIM=.FALSE.,
GARDE_XISH=.FALSE.,
X0=0.70d0,
Y0=0.28d0,
ZSX0=0.d0
/
&NL_CONV
NOM_CONV='conv_jmj',
ALPHA=1.8d0,
OVSHTS=0.d0,
OVSHTI=0.d0,
JPZ=.FALSE.,
CPTURB=0.d0,
LEDOUX=.FALSE.
/
&NL_DIFF
DIFFUSION=.FALSE.,
NOM_DIFFM='diffm_mp',
NOM_DIFFT='difft_nu',
D_TURB=10.d0,
RE_NU=1.d0,
NOM_FRAD='no_frad'
/
&NL_ROT
ROT_SOLID=.TRUE.,
W_ROT=0.d0,
UNIT='jours',
NOM_DIFFW='diffw_mpz',
NOM_PERTW='pertw_0',
P_PERTW=0.d0
/
&NL_ETAT
NOM_ETAT='etat_eff',
F_EOS='eos_opal_250.bin',7*' '
/
&NL_OPA
NOM_OPA='opa_gong',
F_OPA='opa_yveline.bin',7*' '
/
&NL_NUC
NOM_NUC='ppcno9',
NOM_NUC_CPL='NACRE',
MITLER=F
/
&NL_ATM
NOM_ATM='lim_atm',
```

```
 NOM_TDETAU='hopf',
 TAU_MAX=10.d0,
 LIM_RO=.TRUE.
 /
```

Lancer `cesam2k.out`

```
cesam2k.out <==

--------------------------
CESAM speaks a bit of english if you include in
the working directory a file named langue with the statements :
&NL_LANGUE
langue='english'
/
cf. aide_mem2k, chapter Personnalisation
---------------------------

Pour arrêter : taper 0 puis RETURN
Pour poursuivre une évolution : taper 1 puis RETURN
Pour initialiser un modèle de ZAMS : taper 2 puis RETURN
Pour initialiser un modèle de PMS : taper 3 puis RETURN
2  <==
 le modèle initial de ZAMS est-il donné en binaire ? o/n
n <==
 entrer le nom du fichier ASCII du modèle initial
 Exemples: m010.zams, m020.zams, m050.zams

m010.zams <==

 CESAM utilise le modèle m010.zams
 entrer l'identificateur du modèle
 Exemple: mon_modele, modele_euler

mon_modele <==

 identificateur des fichiers du modèle : mon_modele

 ***************************************************************

 MODELE DE STRUCTURE INTERNE calculé par CESAM2k version V1.1.14

 ***************************************************************

 .....                     .....                     .....
   .....                     .....                     .....
      .....                     .....                     .....
```

```
------- Restitution de l'atmosphère (fin) ------
*********
âge= 1.000E+02, LogTeff= 3.778E+00, LogL/Lsol=-5.073E-02, LogR/Rsol=-5.764E-02
Log g= 4.553E+00, Pc= 1.723E+17, Tc= 1.376E+07, Roc= 9.415E+01, Xc= 6.915E-01
en. PP= 97%, en. CNO=  2%, en. 3 alpha=  0%, en. grav=  0%, Yc= 2.879E-01
Var. rel. de masse :  0.000E+00, M*= 1.000E+00Msol, modèle de la série principale
*********

Fin d'évolution avec CESAM2k version : V1.1.14
Sortie car agemax atteint

 .....                       .....                       .....
   .....                       .....                       .....
      .....                       .....                       .....
nom du fichier du modèle d'atmosphère en binaire : mon_modele_B.atm
Le code d'évolution stellaire CESAM a été élaboré
dans le cadre du Groupement de Recherche Structure Interne
des Etoiles et des Planètes Géantes. Si son utilisation
vous a donné satisfaction, le but poursuivi par tous
ceux qui y ont contribué aura été atteint.

P.Morel, ON. Décembre 1989, CESAM1
P.Morel, OCA. Octobre 1991, CESAM2
P.Morel, OCA. Avril 1993, CESAM3
P.Morel, OCA. Décembre 1997, CESAM4
P.Morel, OCA. Décembre 2002, CESAM5
P.Morel, B.Pichon OCA. Septembre 2003, CESAM2k
P.Morel, OCA, Y.Lebreton, MJo Goupil OBSPM, Février 2004, Version anglaise
P.Morel, OCA, A.Moya, OBSPM, Mars 2005, diffusion du moment cinétique

***********************************

 Type <RETURN> for next page:
```

Le test de fonctionnement est terminé.

Lorsque le dessin "on line" est effectué, suivant la définition de l'écran utilisé, il arrive que les cadres soient tronqués et/ou n'utilisent pas la surface disponible de façon satisfaisante. Il y a lieu d'adapter les dimensions des cadres dans les routines `des_m` et `des_r` du sous directory `SOURCE`, *cf.* § 1.5 (Page 4). On recherchera les valeurs optimales en utilisant le fichier `device` de personnalisation du dessin, *cf.* § 4.10 (Page 47).

2.2 Mise en forme des données physiques

CESAM2*k* utilise les données physiques tabulées *i.e.* EOS, opacités *etc...* sous forme de fichiers binaires qu'il faut créer à partir des données `ASCII` du sous-directory `SUN_STAR_DATA` dans lequel on se placera.

PB Il est possible, ce qui est évidemment déconseillé, mais parfois incontournable, d'effectuer des calculs avec des tables d'opacité ou d'équation d'état, dont les abondances en éléments lourds Z diffèrent, ou qui ne correspondent pas à la valeur réelle de Z.

Les fichiers binaires de données créés pour une version précédente de CESAM ***sont utilisables avec*** CESAM2**k**.

2.2.1 Opacités "Yveline"

Décompresser les fichiers `opa_yveline.data.gz` et `ascii2bin_opa.f.gz` :

`gunzip opa_yveline.data.gz ascii2bin_opa.f.gz`

Effectuer la transformation ASCII ==> binaire, en exécutant[1] le programme `ascii2bin_opa` :

`exe2k ascii2bin_opa`

A la question :

```
traduction ASCII ---> binaire (o/n?)
o <==
```

on obtient :

```
ASCII ---> binaire
enter the name of the existing ASCII opacity table: opa_yveline.data
```

répondre :

```
opa_yveline.data <==
```

on obtient :

```
enter the name of the binary table: opa_yveline.bin
```

répondre :

```
opa_yveline.bin <==
```

à la question suivante répondre c, éventuellement recompresser le fichier en ASCII et supprimer l'exécutable. On obtient :

```
PAUSE: OK c'est fait
pour arreter entrer q, poursuivre entrer c
c <==
```

Recompresser les données ASCII : `gzip opa_yveline.data <==` et effacer l'exécutable : `rm ascii2bin_opa.out <==`.
On trouve un complément d'information dans le fichier `extract_opa_yveline_explik`.

[1] Avec F95 il est pratique d'utiliser les procédures du sous-directory `SCRIPTS`.

2.2.2 Opacités OPAL pour $Z > 0.1$

Les opacités OPAL standard n'existent que pour une abondance des métaux inférieure à $Z < 0.1$. Lors de la combustion de l'hélium, dès que l'abondance d'hélium devient inférieure à $Y < 0.9$, `opa_yveline` fait appel au package `z14xcotrin21` d'Arnold I. Boothroyd. Diverses routines permettent de tenir compte de la nucléosynthèse des éléments lourds, en introduisant des corrections aux opacités OPAL. Bien que son implantation dans CESAM2k ne recouvre pas toutes les situations envisagées, son ***USAGE est DELICAT***. Lors de la première utilisation, le système peut demander de décompresser des routines du sous-directory `SUN_STAR_DATA`.

2.2.3 Opacités HOUDEK

Package de calcul de l'opacité OPAL, Iglesias & Rogers (1991), par les routines d'interpolation par splines birationnelles de Houdek & Rogl (1996) ; il est exploité par la routine externe `opa_houdek9`. Ce package, de nom `opint_v9f.tar.gz` peut être obtenu "via anonymous ftp from ftp.ast.cam.ac.uk in the sub-dir `pub/hg/`".

Il est constitué du fichier `opint_v9f.tar.gz` du sous-directory `SUN_STAR_DATA` qu'il faut décompresser et détarer :

```
gunzip opint_v9f.tar.gz <==
tar -xvf opint_v9f.tar <==
v9/
v9/.make.state
v9/.nse_depinfo
v9/.sbinit
v9/Makefile
v9/OPINTPATH_92
v9/OPINTPATH_95
v9/README
v9/Release_history
v9/a2b95.f

............................

v9/rksuite/Makefile
v9/rksuite/chkfl.f
v9/rksuite/.make.state
v9/alex94bext.f
v9/dopdalex94.f
v9/OPINTPATH_AX
v9/exakop95.f
```

Il est prudent de recompresser le fichier `opint_v9f.tar`. En suivant les instructions du README, éditer le makefile du directory `v9` et adapter le compilateur et l'optimiseur à la machine UNIX dont on dispose.
Exécuter le MAKEFILE : `make`.
Dans le fichier `v9/OPINPATH_AX`, remplacer les "./" par les chemins `./v9/opal95/opal95e.bin`, par exemple :

/home/bilou/SUN_STAR_DATA/v9/opal95/opal95e.bin
faire de même pour v9/OPINPATH_91 et v9/OPINPATH_95.

PB Dans la routine d'appel opa_houdek9, le chemin d'accès aux données : tabnam=TRIM(nom_chemin)//TRIM(f_opa(2)) fait appel au **second** fichier f_opa(2) d'opacité, ainsi qu'il est codé dans la NAMELIST NL_OPA du test du § 3.3 (Page 22).
Définir les paramètres iorder et imode en se référant à la notice constituée par le fichier ./doc/notes.ps du directory v9. Dans le script exe2k indiquer au LINK le chemin d'accès à la bibliothèque libopint.a.

PB Pour l'utilisation de ces opacités, dans le module mod_opa et la routine opa, décommenter les instructions concernant opa_houdek9 et reformer l'exécutable cesam2k.out, *cf.* § 1.6 (Page 4).

PB L'implantation de opa_houdek9 est **informatiquement délicate**.

2.2.3.1 Implantation simplifiée de B.Pichon

Se placer dans le sous-directory HOUDEK, exécuter successivement les procédures :

1. build_libopint
2. exebin_alex
3. exebin_opal
4. exebin_opal_alex

Amenager les fichiers OPINPATH

2.2.4 Equation d'état OPAL

2.2.4.1 OPAL 1991

L'équation d'état OPAL, Iglesias & Rogers (1991), est appelée par la routine etat_opal. La source et les tables d'équation d'état ont été reprises du site WEB ftp ://www-phys.llnl.gov/pub/opal/eos/, où l'on trouve un "README" expliquant la marche à suivre pour construire une table ASCII d'équation d'état pour une valeur de Z fixée. On transforme ensuite cette table ASCII en fichier binaire.

PB Certains des sous-programmes du package original d'OPAL ont été modifiés pour l'exploitation avec CESAM2*k* ; en particulier, le blockdata a été supprimé.

Les fichiers ASCII compressés peos*.gz contiennent les données pour l'équation d'état OPAL. Se placer dans le sous-directory SUN_STAR_DATA, décompresser ces fichiers : gunzip peos*.

Exécuter le programme ZFSinterppeos après en avoir décompressé la source : ZFSinterppeos.f.gz

```
exe2k ZFSinterppeos <==
Encountered 0 errors, 0 warnings in file ZFSinterppeos.f.
type Z:
0.02 <==
```

entrer la valeur de Z, fraction de masse de l'abondance des métaux ; on obtient :

```
          0.020
   0.0000   0.0000            3.0000000                 4.0036974
   0.0000   0.0200            3.0301077                 4.0629535
   0.0000   0.0400            3.0611377                 4.1251554
   0.2000   0.0000            2.5018048                 2.5114787
   0.2000   0.0200            2.5158978                 2.5350964
   0.2000   0.0400            2.5302570                 2.5591605
   0.4000   0.0000            2.2741616                 1.8296305
   0.4000   0.0200            2.2828465                 1.8421309
   0.4000   0.0400            2.2916493                 1.8548033
   0.6000   0.0000            2.1437435                 1.4389958
   0.6000   0.0200            2.1498623                 1.4467140
   0.6000   0.0400            2.1560483                 1.4545176
   0.8000   0.0000            2.0592246                 1.1858406
   0.8000   0.0200            2.0638888                 1.1910768
   0.8000   0.0400            2.0685947                 1.1963595
```

Le fichier ASCII `EOSdata` occupant environ 9Mo a été créé.
Editer la source du programme `opal_ascii_bin`, et donner au fichier binaire le nom désiré, par exemple :

```
.....................................

     CLOSE(unit=60)   <===   ligne 117 de opal_ascii_bin.f

c    f_eos='eos_opal_190.bin'
     f_eos='eos_opal_195.bin'
c    f_eos='eos_opal_330.bin'
c    f_eos='eos_opal_180.bin'
c    f_eos='eos_opal_170.bin'
c    f_eos='eos_opal_130.bin'
c    f_eos='eos_opal_050.bin'

     WRITE(*,6)f_eos
6    FORMAT('debut ecriture sur le fichier binaire: ',a50)
     CLOSE(unit=60)

...........................................
```

Exécuter le programme `opal_ascii_bin` ainsi modifié :

```
exe2k opal_ascii_bin <==
Encountered 0 errors, 0 warnings in file opal_ASCII_bin.f.
donnees prises dans le fichier: EOSdata
lecture, et c'est long, de ce fichier
```

```
 fin de lecture des tables OPAL
 debut ecriture sur le fichier binaire: eos_opal_195.bin
 fin ecriture sur le fichier binaire
 donnees prises dans le fichier binaire: eos_opal_195.bin
 lecture, des donnees EOS opal, et c'est long
 fin de lecture des donnees EOS opal en binaire
 test relecture effectue
```

Supprimer les modules exécutables :

```
rm *.out <==
rm: remove regular file ‘ZFSinterppeos.out’? y <==
rm: remove regular file ‘opal_ascii_bin.out’? y <==
```

Enfin, recompresser les fichiers ASCII, et supprimer les fichiers `EOSdata` et `fort.2` désormais inutiles :

```
gzip peos* <==
rm EOSdata <==
rm: remove regular file ‘EOSdata’? y <==
rm fort.2 <==
rm: remove regular file ‘fort.2’? y <==
```

2.2.4.2 OPAL 2001

Les équations d'état 2001 offrent une alternative pour la métallicité : ou bien, comme dans la version 1991, on crée une table pour une métallicité donnée, et l'abondance d'hydrogène doit être dans l'intervalle $X \in [0, 0.8]$, ou bien $X \in [0, 1]$ mais $Z = 0$, *cf.* § 3.7 (Page 27).

– Avec $Z \neq 0$, se placer dans le sous-directory `SUN_STAR_DATA`, et y créer le fichier d'équation d'état correspondant à la métallicité requise en exécutant le programme `Z_interp_IEOS` :

```
 bilou@port-morel% exe2k Z_interp_IEOS
 Encountered 0 errors, 0 warnings in file Z_interp_IEOS.f.
 type Z:  0.025  <===== entrer la valeur de Z
    0.025000000
 bilou@port-morel%
```

Il y a création d'un fichier ACSII compressé de nom `EOSdata.gz`. Transformer ce fichier en binaire, en lui donnant le nom avec lequel il sera identifié par la suite. Pour ce faire, entrer ce nom dans le programme `opalZ_ascii_bin` (vers la ligne 135) :

```
c f_eos='eos_opal_190.bin'
c f_eos='eos_opal_195.bin'
f_eos='eos_opal_250.bin'  <========
c f_eos='eos_opal_330.bin'
c f_eos='eos_opal_180.bin'
```

puis exécuter ce programme qui comporte un test de lecture :

```
bilou@port-morel% exe2k opalZ_ascii_bin
Encountered 0 errors, 0 warnings in file opalZ_ascii_bin.f.
```

```
Fichier de données inconnu : EOSdata
décompression du fichier ASCII de données : EOSdata.gz
données ASCII prises dans le fichier: EOSdata
lecture, et c'est long, de ce fichier
 fin de lecture des tables OPAL
début des écritures sur le fichier binaire: eos_opal_250.bin
fin des écritures en binaire, test de relecture
 données prises dans le fichier binaire: eos_opal_250.bin
 décompression du fichier
 lecture, des données EOS opal, et c'est long
recompression du fichier binaire eos_opal_250.bin
 fin de lecture des données EOS opal en binaire
 test de relecture réussi
bilou@port-morel%
```

Eventuellement supprimer `EOSdata.gz` :

```
 rm EOSdata.gz
```

– Avec $Z = 0$, l'interpolation utilise le fichier `EOSdata_H-He` qu'il suffit de transformer en binaire en exécutant le programme `opalX_ascii_bin` du sous-directory `SUN_STAR_DATA`.

2.2.5 Equation d'état MHD

Package de l'équation d'état MHD, Mihalas et al. (1988), appelé par la routine `etat_mhd`. Ce package a été fourni par W. Däppen. Les 8 tables `mhd1_oc_tau.tab` ... `mhd8_oc_tau.tab` qui se trouvent sur le serveur anonyme usc.edu, dans le répertoire `pub/astro-physics/mhd-oc-tau/wd-evo`, sont disposées dans le sous-directory `SUN_STAR_DATA` sous les noms `mhd1.tab.gz` ... `mhd8.tab.gz`. Se placer dans le sous-directory `SUN_STAR_DATA`, décompresser ces tables en ASCII : `gzip mhd* <==` et la source du programme `fmttob` : `gzip fmttob.f.gz`. Transformer les tables en binaire en exécutant `fmttob <==`

```
exe2k fmttob
Encountered 0 errors, 0 warnings in file fmttob.f.
 decompression

Press Enter to Continue. <==

 enter main
 ZAMS-type: filename of input (formatted) table #  1
 enter filename of output (unformatted) table #  1
 ZAMS-type: filename of input (formatted) table #  2
 enter filename of output (unformatted) table #  2
 ZAMS-type: filename of input (formatted) table #  3
 enter filename of output (unformatted) table #  3
 center-type: filename of input (formatted) table #  4

 ..................................
```

```
     4.002600       7.7222100E-02   0.2299999

     12.01100       4.2665000E-04   3.8132454E-03

     14.00670       1.0717000E-04   1.1169993E-03

     15.99940       9.1215000E-04   1.0859599E-02

     20.17900       2.8039000E-04   4.2102317E-03

 mean molecular weight =    1.2455340


bilou@port-morel%
```

Supprimer les exécutables :

```
rm *.out <==
rm: remove regular file fmttob.out ?  y <==
```

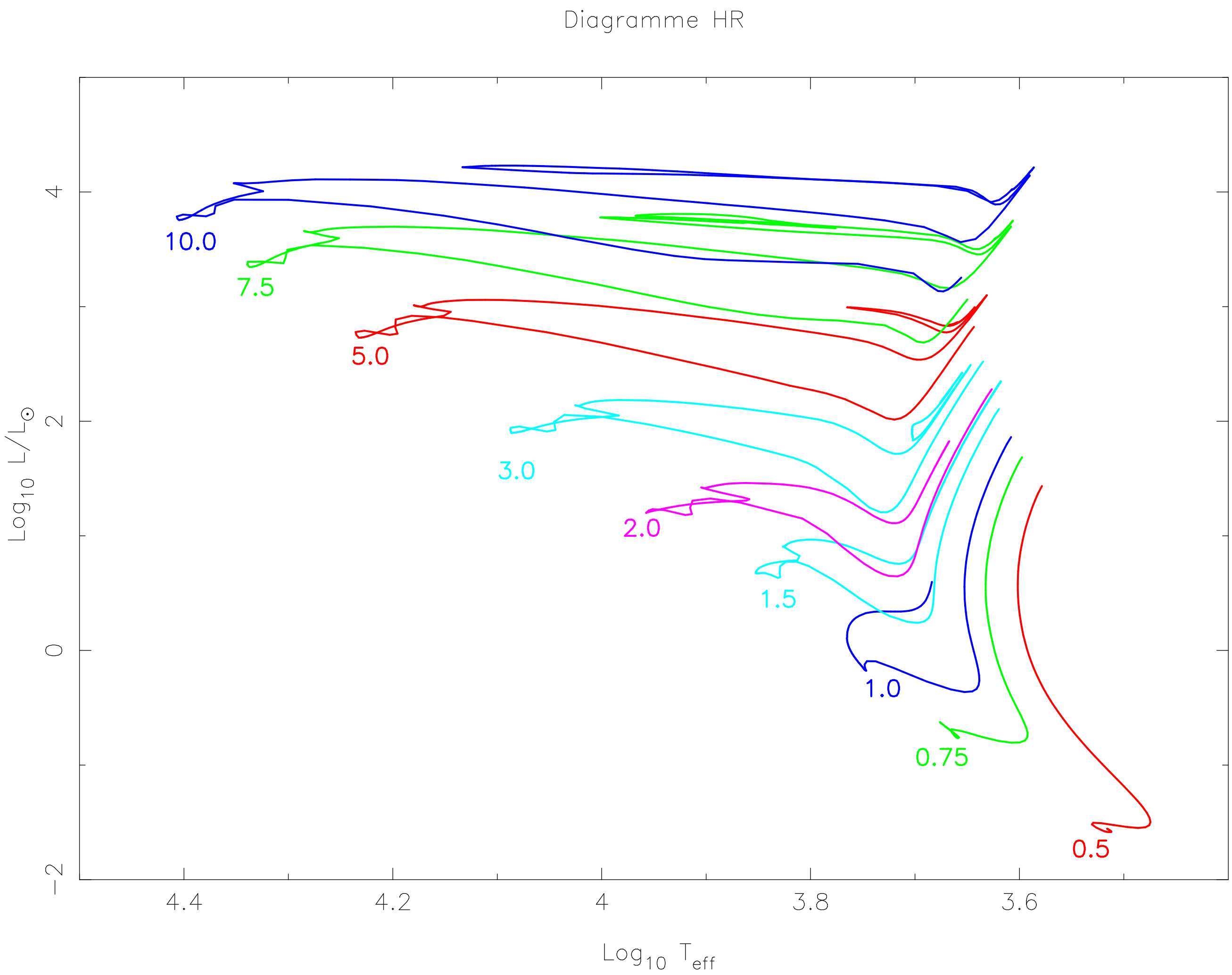

FIG. 2.1 – Tracés évolutifs incluant la pré-séquence principale pour des étoiles de $0.5M_\odot$ à $10M_\odot$. Ces tracés ont été obtenus avec le fichier de données du § 3.3 (Page 22).

Chapitre 3

Exploitation

Va despacio, estoy de prisa.
Proverbe Mexicain.
(Je vais doucement car je suis pressé)

3.1 Conseils pour l'exploitation

Un code numérique, aussi bien fait soit-il, ne peut être "clefs en mains". Pour entamer une exploitation, on doit toujours être guidé par la remarque triviale :

Un programme qui ne fonctionne pas ou mal pour des cas simples
ne fonctionnera pas ou encore plus mal pour des cas compliqués.

Un code de structure interne n'échappe, hélas pas, à cette règle. A cet effet, bien que CESAM2*k* ait prévu différents types de "précision" standard, il est presque toujours nécessaire, à l'aide de "réglages", d'optimiser les paramètres numériques en fonction des conditions rencontrées au cours du calcul. Aborder un problème en introduisant *ad initio* toute la complexité que l'on désire atteindre est une perte de temps, car il faudra très rapidement reprendre des cas simples, afin d'acquérir une expérience sur la façon dont l'évolution se déroule. Il est conseillé de "dégrossir" le problème en utilisant des options numériques et physiques robustes et efficaces, afin d'apprécier rapidement "ce qui se passe", puis d'introduire progressivement la complexité souhaitée.

$\boxed{\textbf{PB}}$ `Les descriptions des routines données plus avant ne sont que succinctes, se reporter aux commentaires des routines pour plus d'informations.`

3.2 Le fichier de données : `mon_modele.don`

Les options physiques à utiliser et les données sont transmises au programme principal par des `NAMELIST` regroupées dans un fichier en ASCII dont le nom a **nécessairement** l'extension `.don`, par exemple : `mon_modele.don`. On a disposé un exemple de ce fichier dans le sous-directory `EXPLOIT`.

– `& NL_CESAM` : regroupe les conditions générales du calcul.

- & `NL_MASS` : regroupe les paramètres physiques relatifs à la variable spatiale.
- & `NL_EVOL` : regroupe les paramètres physiques relatifs à la variable temporelle.
- & `NL_CHIM` : regroupe les paramètres physiques relatifs à la composition chimique.
- & `NL_CONV` : regroupe les paramètres physiques relatifs à la convection.
- & `NL_DIFF` : regroupe les paramètres physiques relatifs à la diffusion.
- & `NL_ROT` : regroupe les paramètres physiques relatifs à la rotation.
- & `NL_ETAT` : regroupe les noms des fichiers, *d'accès externe*, éventuellement nécessaires au calcul de l'équation d'état.
- & `NL_OPA` : regroupe les noms des fichiers, *d'accès externe*, éventuellement nécessaires au calcul de l'opacité.
- & `NL_NUC` : définit des options relatives au calcul des taux de réactions thermonucléaires.
- & `NL_ATM` : regroupe les paramètres physiques relatifs à la restitution de l'atmosphère.

3.3 Exemple de fichier de données

Cet exemple est volontairement différent de celui utilisé au § 2.1 (Page 9).

```
&NL_CESAM
NOM_CHEMIN='~/SUN_STAR_DATA/',
NOM_CTES='ctes_94',
NOM_DES='des_r',
NOM_OUTPUT='no_output',
N_MAX=2000,
PRECISION='pr'
/
&NL_MASS
MTOT=2.d0,
NOM_PERTM='pertm_ext',
MDOT=0.d0
/
&NL_EVOL
AGEMAX=5.d3,
ARRET='else',
DTLIST=1.d10,
LOG_TEFF=10.d0,
NB_MAX_MODELES=300,
HE_CORE=0.1d0,
T_STOP=5.d7,
X_STOP=0.1d0
/
&NL_CHIM
GRILLE_FIXE=.FALSE.,
NOM_ABON='solaire_gn',
MODIF_CHIM=.FALSE.,
```

```
GARDE_XISH=.FALSE.,
X0=0.70,
Y0=0.28,
ZSX0=0.0d0
/
&NL_CONV
NOM_CONV='conv_jmj',
ALPHA=1.8d0,
OVSHTS=0.d0,
OVSHTI=0.d0,
JPZ=F,
CPTURB=0.d0,
LEDOUX=F
/
&NL_DIFF
DIFFUSION=.TRUE.,
NOM_DIFFM='diffm_mp',
NOM_DIFFT='difft_nu',
D_TURB=10.d0,
RE_NU=1.d0,
NOM_FRAD='no_frad'
/
&NL_ROT
ROT_SOLID=.TRUE.,
W_ROT=0.d0,
UNIT='kms/s',
NOM_DIFFW='diffw_mpz',
NOM_PERTW='pertw_0'
/
&NL_ETAT
NOM_ETAT='etat_eff',
F_EOS='eos_opal_250.bin',7*' '
/
&NL_OPA
NOM_OPA='opa_yveline',
F_OPA='opa_yveline.bin','v9/OPINTPATH_AX',6*' '
/
&NL_NUC
NOM_NUC='ppcno9',
NOM_NUC_CPL='NACRE',
MITLER=F
/
&NL_ATM
NOM_ATM='lim_atm',
NOM_TDETAU='hopf',
TAU_MAX=50.d0,
LIM_RO=.TRUE.
```

```
/
```

PB `Les noms de routines sont` ***case sensitive***`, par exemple, il faut coder NACRE et non nacre.`

3.4 Arguments de la NAMELIST `NL_CESAM`

- `NOM_CHEMIN='~/SUN_STAR_DATA/'` : chemin et nom du répertoire où sont disposées les données physiques, les tables d'opacité par exemple.
- `NOM_CTES` : indique le nom de la routine d'initialisation des principales constantes de physique à utiliser :
  - `ctes_85` : constantes de physique proposées par GONG,
  - `ctes_94` : constantes de physique récentes.
- `NOM_DES` : nom de la routine de dessin à utiliser :
  - `des_m` : dessin en fonction de la masse, utilise PGPLOT,
  - `des_r` : dessin en fonction du rayon, utilise PGPLOT,
  - `no_des` : pas de dessin, à utiliser ,en particulier, si on ne dispose pas du logiciel PGPLOT, voir § 1.4 (Page 3),
  - `zoom` : forme personnalisée du dessin décrite au § 4.9 (Page 45), pratique lorsqu'on désire atteindre des conditions particulières, par exemple une localisation dans le diagramme HR.
- `NOM_OUTPUT` : type de fichier ASCII à générer à l'issue du calcul :
  - `osc_adia` : génération du fichier ASCII de nom `mon_modele-ad.osc` pour le calcul des oscillations adiabatiques et les dessins,
  - `all_adia` : génération des fichiers ASCII, pour le calcul des oscillations adiabatiques et les dessins, de ***TOUS*** les modèles. Ces fichiers ont pour dénomination `mon_modele-ad`***nnnn***`.osc`, *nnnn* est le numéro du modèle,
  - `osc_invers` : génération des fichier ASCII de nom `mon_modele-inv.osc` pour les inversions,
  - `all_invers` : génération de ***TOUS*** les fichiers ASCII de noms `mon_modele-inv`***nnnn***`.osc` pour les inversions,
  - `osc_nadia` : génération du fichier ASCII de nom `mon_modele-nad.osc` pour le calcul des oscillations non adiabatiques,
  - `all_nadia` : génération de ***TOUS*** les fichiers ASCII de noms `mon_modele-nad`***nnnn***`.osc` pour le calcul des oscillations non adiabatiques, *nnnn* étant le numéro du modèle,
  - `no_output` : il n'y a pas de génération de fichier ASCII,
  - `ascii` : forme personnalisée du fichier ASCII `mon_modele-ascii` décrite au § 4.7 (Page 42),
  - `all_ascii` : génération de ***TOUS*** les fichiers ASCII de forme personnalisée `mon_modele-ascii`***nnnn***, *nnnn* étant le numéro du modèle.
- `N_MAX=1000` : le nombre maximum de couches[1] est défini par `MAX(N_MAX,nnnn)`, `nnnn` est un nombre maximal de couches fixé dans la routine `cesam` suivant le

[1] La nature des paramètres numériques et logiques est identifiable par les valeurs données à titre indicatif.

niveau de précision requis ; parallèlement, un nombre minimal de couches est fixé à 300.
- PRECISION='sp' : niveau de précision requis. On trouve dans la Table 3.1 (Page 26) les valeurs des paramètres numériques utilisées suivant les différentes options définies dans la routine cesam. Les valeurs conseillées sont :
  - 'np' (précision normale) pour se donner une idée générale d'une évolution, les coefficients pour la diffusion du moment cinétique sont calculés avec la routine coeff_rota_z16,
  - 'pr' (précision réaliste) pour une évolution normale, , les coefficients pour la diffusion du moment cinétique sont calculés avec la routine coeff_rota_z16,
  - 'sp' (super précision) et 'sa' (précision solaire) si on recherche une précision extrême. Avec 'sa' les derniers modèles d'une évolution sont calculés avec le nombre maximum de couches N_MAX ; les coefficients pour la diffusion du moment cinétique sont calculés avec la routine coeff_rota_saha,

  - 'mx' (maximum de couches) identique à 'sp' le calcul se faisant avec un nombre ***fixe*** de couches, égal à N_MAX ; les coefficients pour la diffusion du moment cinétique sont calculés avec la routine coeff_rota_saha.

  Les constantes de répartition sont fixées indépendamment du type de précision. Les valeurs sont celles retenues dans CESAM4 à savoir : ctel=0, ctep=-1, ctem=15, cter=0, ctet=0. Pour utiliser des paramètres différents *cf.* § 4.8 (Page 43).

3.5 Arguments de la NAMELIST NL_MASS

- MTOT=1.d0 : masse totale initiale, en $M_\odot$,
- NOM_PERTM : nom de la routine de perte de masse :
  - pertm_ext : perte/gain de masse externe linéaire en fonction du temps.
  - pertm_msol : perte/gain de masse externe linéaire en fonction du temps, la masse de l'étoile restant au moins égale à 1 $M_\odot$,
  - pertm_tot : perte/gain de masse externe, linéaire en fonction du temps et tenant compte de la perte de masse due aux réactions thermonucléaires ($E = mc^2$),
  - pertm_waldron : perte de masse empirique de Waldron cité dans A.Ap 229 (1990) 469-474.
- MDOT=1.d-14 : taux de perte/gain de masse, en $M_\odot$/an, ce taux est positif pour un gain de masse, négatif pour une perte de masse. Dans le cas standard, la composition chimique de l'apport/retrait de masse est celle de la couche la plus externe du modèle. Voir § 4.5 (Page 40) comment personnaliser, si besoin, cette composition chimique.

3.6 Arguments de la NAMELIST NL_EVOL

- AGEMAX=4.5d4 : âge maximal à atteindre en million d'années,
- ARRET='post' : arrêt sur la ZAMS ('zams'), à la fin de la ZAMS ('post'), au début de la combustion de l'hélium ('cohe') ou du carbone ('coca') ou autre

Tab. 3.1 – Valeurs des paramètres numériques suivant les diverses options de précision prévues. Les valeurs de ces paramètres peuvent être adaptées au type d'évolution, en utilisant l'option `'rg'`, *cf.* § 4.8 (Page 43) ; on utilise alors les paramètres indiqués dans le fichier de nom `reglages` placé dans l'environnement. Les dénominations s'interprètent de la façon suivante : `sp` : super précision, `pr` : précision réaliste, `lm` : pour petites masses (low mass), `sa` : pour modèles avec précision solaire (solar accuracy), `np` : pour modèles de précision normale, `co` : pour modèles destinés à l'exploitation des résultats du satellite CoROT. La signification des différents paramètres numériques est explicitée au § 4.8 (Page 43) et dans la routine `cesam` du sous-directory `SOURCE`, lignes 741-766 ; d'autres options sont prévues, certaines utilisent les abscisses eulériennes ; elles ne sont pas détaillées ici.

	sp	pr	lm	sa	np	co
m_qs	2	2	2	2	1	2
m_ch	3	3	3	4	2	3
psi0	0.05	0.05	0.08	0.02	0.08	0.04
m_rot	3	3	3	4	2	3
m_tds	2	2	2	2	2	2
m_ptm	2	2	2	2	2	2
ordre	2	2	2	2	1	2
precix	10^{-4}	10^{-3}	10^{-4}	10^{-5}	0.005	10^{-4}
precit	0.05	0.15	0.08	0.02	0.3	0.05
ro_test	0.1	0.1	0.1	0.1	0.1	0.1
d_grav	0.5	0.5	0.5	0.5	0.5	0.5
loc_zc	10^{-4}	10^{-3}	10^{-4}	10^{-5}	0.005	10^{-4}
dtmax	50	200	300	50	300	200
ini0	5	4	5	5	4	5
n_atm	100	75	75	100	50	100
kipp	.FALSE.	.FALSE.	.TRUE.	.FALSE.	.TRUE.	.FALSE.
en_masse	.TRUE.	.TRUE.	.TRUE.	.TRUE.	.TRUE.	.TRUE.
mvt_dis	.TRUE.	.TRUE.	.TRUE.	.TRUE.	.FALSE.	.TRUE.
n_max	1000	1000	1000	2000	1000	2000
dn_fixe	0.05	0.05	0.1	0.05	0.05	0.05

('else').

- DTLIST=1.d3 : intervalle de temps minimum, en million d'années, séparant deux listes détaillées du modèle (fichier mon_modele.lis),
- LOG_TEFF=-3.7d0 : arrêt si cette valeur de $\log_{10} T_{\rm eff}$ est traversée, dans le sens croissant si log_teff est positif, dans le sens décroissant si log_teff est négatif,
- NB_MAX_MODELES=500 : arrêt après avoir calculé le nombre de modèles indiqué, éventuellement 0 ; avec NB_MAX_MODELES < 0, ***tous*** les fichiers binaires seront écrits dans l'environnement, leurs noms mon_modele*nnnn*_B.*** comportant le numéro du modèle et l'extension d'identification, .hom, .rep, *etc...* . Le numéro 0000 est affecté aux modèles d'initialisation de séquence principale d'âge zéro ou de pré-séquence principale ; en cas de reprise, les numéros reprennent à partir de celui du modèle repris.
- HE_CORE=0.1d0 : arrêt, dès que la masse du cœur d'hélium atteint la valeur he_core ($M_\odot$),
- T_STOP=1.d7 : arrêt, si au centre, la température dépasse cette valeur,
- X_STOP=-0.1d0 : arrêt si cette valeur de X, au centre, est traversée.

3.7 Arguments de la NAMELIST NL_CHIM

- GRILLE_FIXE=.TRUE. : on utilisera une grille fixe pour la composition chimique et la diffusion du moment cinétique,
- NOM_ABON : nom de la mixture initiale[2] :
  - enhan_al : abondances α-enhanced de Allard,
  - enhan_cha : abondances α-enhanced de Chaboyer,
  - enhan_w : abondances α-enhanced de Weiss,
  - meteorites_ag : abondances météoritiques de Anders & Grevesse 89,
  - meteorites_gs : abondances météoritiques de Grevesse & Sauval 98,
  - mixture : forme personnalisée de la compilation d'abondances décrite au § 4.2 (Page 38),
  - solaire_gn : mixture solaire de Grevesse & Noels 93,
  - solaire_gs : mixture solaire de Grevesse & Sauval 98.

  Ces abondances sont initialisées dans la routine abon_ini.
- MODIF_CHIM=.FALSE. : si, dans l'environnement, existent des fichiers des types modif_mix, rap_iso, planet ou vent qui permettent, respectivement, de modifier la composition chimique initiale, *cf.* § 4.4 (Page 40), les rapports isotopiques, *cf.* § 4.3 (Page 39), la composition chimique des planétoïdes, *cf.* § 4.6 (Page 41) ou celle du vent, *cf.* § 4.5 (Page 40), CESAM2*k* invitera à supprimer ces fichiers avant d'effectuer les calculs. Cette disposition ***à but sécuritaire*** a pour fin d'éviter de prendre en compte des fichiers qui, par mégarde, n'auraient pas été soustraits de l'environnement. Il convient de coder MODIF_CHIM=.TRUE. pour tenir compte des dispositions offertes par la présence de ces fichiers.
- GARDE_XISH=.FALSE. : ce paramètre fait l'objet du paragraphe suivant,
- X0=0.7 : abondance initiale, par unité de masse, de H,
- Y0=0.28 : abondance initiale, par unité de masse, de He,

[2] *cf.* équation d'état OPAL : WEB site http ://www-phys.llnl.gov/V_Div/OPAL/opal.html.

- `ZSX0=0.d0` : valeur initiale de Z/X,
  - Les abondances du modèle initial sont déterminées à partir des quantités `X0`, `Y0` et `ZSX0` ; en raison de la relation $1 = X + Y + Z$, seulement 2 de ces 3 quantités sont à prendre en considération. La valeur initiale de Y utilisée dans le calcul est ***toujours*** `Y0` ; elle est fractionnée en les divers isotopes retenus pour décrire l'évolution de l'hélium,
  - si `ZSX0` $\leq$ `0`, l'abondance initiale d'hydrogène est initialisée à `X0`, $Z = 1 - X - Y$ est déduit de `X0` et `Y0`,
  - sinon, *i.e.* `ZSX0` $>$ `0`, l'abondance initiale d'hydrogène est déduite de `Y0` et de `ZSX0`, la valeur indiquée pour `X0` est ignorée.

La valeur initiale de X utilisée dans le calcul est fractionnée en les divers isotopes retenus pour décrire l'évolution de l'hydrogène.
Avec diffusion et/ou après un premier dredge-up, la composition chimique des couches extérieures, *i.e.* celle observée, diffère de la composition chimique initiale. Il y a lieu d'ajuster la valeur initiale de la composition chimique, afin d'obtenir celle observée à l'issue d'une évolution. Les observations portant sur les rapports métal/H, ou encore [Fe/H], à l'initiative de F.Thévenin, CESAM2*k* offre divers outils pour réaliser cet ajustement :

- Conserver le rapport Z/X de la mixture initiale, au lieu de `ZSX0`, objet du § 3.7.1 (Page 28),
- utiliser une mixture originale, *cf.* § 4.2 (Page 38),
- modifier les rapports isotopiques, *cf.* § 4.3 (Page 39),
- modifier les rapports d'abondances de la mixture initiale, *cf.* § 4.4 (Page 40).

3.7.1 Alternative : conservation métal/H ou métal/Z

Pour une mixture initiale, éventuellement modifiée ou non, *cf.* § 4.4 (Page 40), l'alternative `GARDE_XISH=.TRUE.` (*respt.* `GARDE_XISH=.FALSE.`) permet de fixer la valeur initiale de Z (*respt.* X) de façon à conserver le rapport `métal/H` (*respt.* `métal/Z`). La valeur initiale de Y est ***TOUJOURS*** la valeur lue dans le fichier de données, *i.e.* `Y0`.

Ainsi, en codant[3] `GARDE_XISH=.TRUE.`, la valeur de Z/X utilisée sera déduite des rapports d'abondances métal/H de la mixture et non pas celle, `ZSX0`, lue dans le fichier de données[4]. Les rapports d'abondances métal/H, seront ceux de la mixture, les rapports métal/H seront différents de ceux de la mixture.

A l'inverse, en codant `GARDE_XISH=.FALSE.`, la valeur `ZSX0` lue dans le fichier de données, sera utilisée. Les rapports métal/Z seront ceux de la mixture[5], les rapports métal/H seront différents de ceux de la mixture. Pour une ***calibration solaire***, pour laquelle on vise à atteindre une valeur fixée de Z/X, en conservant les rapports métal/Z initiaux, il convient de coder `garde_xish=.FALSE.`

$\boxed{\text{PB}}$ `L'utilisation de GARDE_XISH=.TRUE.` ***EST DÉLICATE***`, il est conseillé de bien vérifier, dans le fichier mon_modele.lis, que les`

[3]De meilleures dénominations seraient, par exemple : `GARDE_XISH_MIX`, `GARDE_XISH_MIX_INI`, `GARDE_ZSX` ou encore `GARDE_ZSX_MIX_INI`,*etc...*

[4]Sauf dans le cas particulier où les valeurs de Y0 et ZSX0 du fichier de données correspondent **exactement** à celles de la mixture.

[5]De façon cohérente avec les tables d'opacités.

valeurs des abondances utilisées correspondent bien à ce que l'on désire.

PB Il faut être conscient que les modifications de la mixture ne seront pas prises en compte, ni dans les opacités, ni dans l'équation d'état.

3.8 Arguments de la NAMELIST NL_CONV

- NOM_CONV : nom de la routine calculant le gradient de température dans les zones convectives.
  - conv_a0 : convection MLT, avec longueur de mélange $l \to 0$ aux limites ZR/ZC,
  - conv_cgm_reza : convection suivant Canuto & Mazitelli avec $l = \alpha H_{\rm p}$, tient compte de la prescription de Bernkopf,
  - conv_cm : convection suivant Canuto & Mazitelli avec $l = \alpha H_{\rm p}$,
  - conv_cm_reza : convection suivant Canuto & Mazitelli avec $l = \alpha H_{\rm p}$, tient compte de δ,
  - conv_jmj : convection MLT avec $l = \alpha H_{\rm p}$.
- alpha=1.7d0 : longueur de mélange,
- ovshts=0.05d0 : coefficient d'overshoot supérieur, une valeur positive affecte le gradient adiabatique à la partie overshootée, une valeur négative affecte le gradient radiatif,
- ovshti=0.1d0 : coefficient d'overshoot inférieur, une valeur positive affecte le gradient adiabatique à la partie overshootée, une valeur négative affecte le gradient radiatif,
- jpz=.FALSE. : utilisation des prescriptions de JpZ,
- cpturb=0.d0 : coefficient de pression turbulente,
- ledoux=.FALSE. : utilisation du critère de Ledoux[6].

PB L'utilisation d'un paramètre de pression turbulente non nul est délicate.

3.9 Arguments de la NAMELIST NL_DIFF

- DIFFUSION=.TRUE. : on tiendra compte de la diffusion microscopique des éléments chimiques, éventuellement de la diffusion du moment cinétique,
- NOM_DIFFM : nom de la routine calculant les coefficients de diffusion microscopique :
  - diffm_br : calcul des coefficients de diffusion microscopique suivant le formalisme de Burgers en tenant compte, éventuellement, des accélérations radiatives,
  - diffm_mp : calcul des coefficients de diffusion microscopique suivant le formalisme simplifié de Michaud & Proffit,
  - diffm_0 : coefficients de diffusion microscopiques nuls.

[6] A n'utiliser qu'avec précautions.

- `NOM_DIFFT` : nom de la routine calculant les coefficients de diffusion turbulente :
  - `difft_nu` : calcul des coefficients de diffusion turbulente incluant la diffusivité radiative.
  - `difft_gab` : calcul des coefficients de diffusion turbulente selon une prescription simple de M.Gabriel.
  - `difft_sun` : coefficients de diffusion turbulente, sous la zone convectivesolaire suivant M. Gabriel 1997, A&A 327, 771.
- `D_TURB=1.d1` : coefficient de diffusion turbulente isotrope,
- `RE_NU=1.d0` : coefficient de diffusivité radiative,
- `NOM_FRAD` : nom de la routine de calcul des accélérations radiatives,
  - `alecian1` : calcul des accélérations radiatives suivant le premier formalisme de G.Alécian,
  - `alecian2` : calcul des accélérations radiatives suivant le second formalisme de G.Alécian,
  - `no_frad` : les accélérations radiatives seront ignorées.

3.10 Arguments de la NAMELIST `NL_ROT`

- `ROT_SOLID = .TRUE.` : la vitesse angulaire locale sera la même en tout point du modèle :
  - `w_rot >= 0` : le moment cinétique est localement conservé, ses variations locales sont transformées en énergie interne,
  - `w_rot < 0` : le moment cinétique est globalement conservé, la vitesse angulaire locale `wrot`, la même en tout point du modèle, variera au cours du temps ; elle sera réajustée au cours des itérations à l'issue du calcul du moment cinétique total, dont on assurera la conservation globale en faisant varier `wrot`.
- `ROT_SOLID = .FALSE.` : il y a diffusion du moment cinétique dans les zones radiatives, mélange dans les zones convectives. Le cas incohérent `rot_solid =.FALSE.` avec `w_rot=0` est éliminé. Il est possible de calculer des modèles **avec** diffusion du moment cinétique mais **sans** diffusion des éléments chimiques, l'intérêt de tels modèles réside uniquement dans la prise en compte de la diminution de la gravité locale due à l'accélération cintrifuge.
- `W_ROT=50.d0` : vitesse angulaire initiale.
- `UNIT` : unité utilisée pour la vitesse angulaire initiale. différents choix sont proposés de façon à s'adapter facilement aux observations :
  - `jours` : période initiale en jours, correspond à la vitesse angulaire initiale de rotation de la couche externe,
  - `kms/s` : vitesse de la couche externe en kilomètres/seconde, le rayon du modèle d'initialisation étant utilisé pour déterminer la vitesse angulaire initiale, il convient d'ajuster ce rayon en reprenant plusieurs fois le calcul du modèle d'âge 0,
  - `rad/s` : la vitesse angulaire initiale est en radians/seconde.
- `NOM_DIFFW` : nom de la routine du calcul des coefficients de diffusion du moment cinétique :
  - `diffw_p03` : coefficients de diffusion du moment cinétique de Palacios et al.

  - diffw_mpz : coefficients de diffusion du moment cinétique de Matis, Palacios & Zahn.
  - diffw_0 : coefficients de diffusion du moment cinétique constants.
- NOM_PERTW : nom de la routine de calcul de perte de moment cinétique,
  - pertw_sch : variation de vitesse angulaire proportionnelle à Ω^3 (Schumanish),
  - pertw_loc : perte de moment cinétique proportionnelle à l'énergie cinétique de rotation locale,
  - pertw_ptm : perte de moment cinétique, conséquence d'une perte de masse,
  - pertw_0 : pas de perte de moment cinétique.
- P_PERTW=-1.d-9 : paramètre de perte de moment cinétique attaché respectivement à chacune des routines précédentes.

3.11 Arguments de la NAMELIST NL_ETAT

- NOM_ETAT : nom de la routine d'équation d'état :
  - etat_ceff : équation d'état de Eggleton et al., avec corrections Coulombiennes (*appel à* etat_eff *en cas de difficulté*).
  - etat_eff : équation d'état de Eggleton et al.(*appel à* etat_gong2 *en cas de difficulté*).
  - etat_gong1 : équation d'état de gong1, seuls H et He sont pris en compte et supposés totalement ionisés.
  - etat_gong2 : équation d'état de gong2, seuls H et He4 sont pris en compte.
  - etat_mhd : équation d'état de Mihalas, Hummer & Dappen (*appel à* etat_eff *en cas de difficulté*) utilise les tables en binaire mhd1.bin ... mhd7.bin construites dans le sous-directory SUN_STAR_DATA, voir § 2.2.5 (Page 18).
  - etat_opal : équation d'état opal 1991, (*appel à* etat_eff *en cas de difficulté*) utilise les tables en binaire eos_opal*.bin construites dans le sous-directory SUN_STAR_DATA, voir § 2.2.4 (Page 15).
  - etat_opalZ : équation d'état de opal 2001, (*appel à* etat_eff *en cas de difficulté*) utilise les tables en binaire eos_opalZ*.bin construites pour **Z fixé** dans le sous-directory SUN_STAR_DATA, *cf.* § 2.2.4 (Page 15). Par rapport à la version 1991, les données de la version 2001 ont été étendues et améliorées, des bugs ont été corrigés. Les données ne permettent pas de dépasser $X = 0.8$. Les fichiers de données 2001 et 1991 sont incompatibles.
  - etat_opalX : équation d'état de opal 2001, se différenciant de etat_opalZ par l'extension du domaine de l'abondance d'hydrogène, $X \in [0, 1]$, dans le but de calculer des modèles sans zone convective externe étendue *avec diffusion*. Le prix à payer est $Z \equiv 0$, ce qui n'est pas très pénalisant, l'équation d'état étant peu sensible à Z. etat_opalX utilise **la** table en binaire eos_opalX construite pour **Z=0 fixé** dans le sous-directory SUN_STAR_DATA, *cf.* § 2.2.4 (Page 15).
- F_EOS='eos_opal_250.bin',7*' ' : noms des fichiers d'équation d'état (au plus 8).

3.12 Arguments de la NAMELIST NL_OPA

- NOM_OPA : nom de la routine de calcul des opacités moyennes de Rosseland :
  - opa_gong : opacités simplifiées (Kramers amélioré).
  - opa_houdek9 : opacités de Houdek version 9, (OPAL+Alexander), interpolation par rational B-spline ; utilise les tables en binaire construites dans le sous-directory SUN_STAR_DATA/V9, voir § 2.2.3 (Page 14)
  - opa_int_zsx : Opacités OPAL93+Kurucz raccord d'Yveline, interpolations linéaires, très utiles pour des tests.
  - opa_opal2.f : Opacités OPAL avec corrections pour $Z > 0.1$ *cf.* § 2.2.2 (Page 14)
  - opa_opalCO : Opacités OPAL avec corrections pour $Z > 0.1$ ne portant que sur C et O.
  - opa_yveline : opacités OPAL+Alexander interpolation et raccord d'Yveline ; utilise les tables en binaire opa_yveline*.bin construites dans le sous directory SUN_STAR_DATA, voir § 2.2.1 (Page 13).
  - opa_yveline_lisse : opacités OPAL+Alexander raccord d'Yveline, interpolation linéaire ou lissage ; utilise les tables en binaire opa_yveline*.bin construites dans le sous-directory SUN_STAR_DATA, voir § 2.2.1 (Page 13).
- F_OPA='opa_yveline.bin',7*' ' : noms des fichiers d'opacité (au plus 8).

3.13 Arguments de la NAMELIST NL_NUC

- NOM_NUC : nom de la routine de calcul des taux des réactions thermonucléaires et d'initialiation de la composition chimique :
  - pp1 : calcul simplifié du cycle PP (ne permet pas la diffusion microscopique).
  - pp3 : réactions PP, 3 éléments H, He3, He4 avec H2, Li7, Be7 à l'équilibre ; a pour vocation les tests de mise au point, permet la diffusion microscopique.
  - ppcno9 : réactions PP+CNO, 9 éléments, H2, Li7, Be7 à l'équilibre.
  - ppcno9Fe : réactions PP+CNO, 9 éléments+Fe, H2, Li7, Be7 à l'équilibre.
  - ppcno10 : réactions PP+CNO, 10 éléments, H2, Be7 à l'équilibre.
  - ppcno10Fe : réactions PP+CNO, 10 éléments+Fe, H2, Be7 à l'équilibre.
  - ppcno10BeBFe : réactions PP+CNO, 10 éléments + Li6, Be9, B11, Fe avec H2 et Be7 à l'équilibre.
  - ppcno11 : réactions PP+CNO, 11 éléments, Be7 à l'équilibre.
  - ppcno12 : réactions PP+CNO, 12 éléments.
  - ppcno12Be : réactions PP+CNO, 12 éléments + Be9.
  - ppcno12Li : réactions PP+CNO, 12 éléments + Li6.
  - ppcno12BeBFe : réactions PP+CNO, 12 éléments + Li6, Be9, B11, Fe.
  - ppcno3a12Ne : réactions PP+CNO+3α, 12 éléments dont Ne22, H2, Li7, Be7 à l'équilibre.
  - ppcno3a9 : réactions PP+CNO+3α, 9 éléments, H2, Li7, Be7 à l'équilibre.
  - ppcno3ac10 : réactions PP+CNO+3α+C, 10 éléments H2, Li7, Be7 à l'équilibre.
- nom_nuc_cpl : nom de la compilation des réactions thermonucléaires :
  - Adelb : Compilation d'Adelberger et al. [1998].

  - Cau-Fow : Compilation de Caughlan & Fowler [1988].
  - NACRE : Compilation de NACRE (Angulo et al. [1999]).
- MITLER=.FALSE. : effet d'écran selon Mitler.

3.14 Arguments de la NAMELIST NL_ATM

- NOM_ATM : nom de la routine de restitution de l'atmosphère :
  - lim_atm : restitution de l'atmosphère à partir d'une loi $T(\tau)$,
  - lim_gong1 : restitution simplifiée de l'atmosphère, cas de GONG1,
  - lim_tau1 : restitution simplifiée de l'atmosphère, atmosphère mono-couche.
- NOM_TDETAU : nom de la loi $T(\tau)$ utilisée pour la restitution de l'atmosphère :

  - edding : loi $T(\tau)$ totalement radiative d'Eddington.
  - hopf : loi $T(\tau)$ totalement radiative de Hopf.
  - K5750 : loi $T(\tau)$ non totalement radiative, dérivée du modèle d'atmosphère solaire de Kurucz, $T_{\rm eff} = 5750\,$K.
  - K5777 : loi $T(\tau)$ non totalement radiative, dérivée du modèle d'atmosphère solaire de Kurucz, $T_{\rm eff} = 5777\,$K.
  - roger00 : loi $T(\tau)$ non totalement radiative, dérivée de modèles d'atmosphère de Kurucz, de métallicité $[\frac{\rm Fe}{\rm H}] = 0.0$.
  - roger02 : loi $T(\tau)$ non totalement radiative, dérivée de modèles d'atmosphère de Kurucz, de métallicité $[\frac{\rm Fe}{\rm H}] = +0.2$.
  - roger05 : loi $T(\tau)$ non totalement radiative, dérivée de modèles d'atmosphère de Kurucz, de métallicité $[\frac{\rm Fe}{\rm H}] = -0.5$.
  - roger10a : loi $T(\tau)$ non totalement radiative, dérivée de modèles d'atmosphère de Kurucz, de métallicité $[\frac{\rm Fe}{\rm H}] = -1.0$ avec éléments α "enhanced".
- TAU_MAX=10.d0 : profondeur optique du raccord avec l'enveloppe,
- LIM_RO=.TRUE. : condition limite externe en densité, sinon en pression.

3.15 Fichiers de données des versions antérieures

CESAM2k accepte d'exploiter les fichiers de données de CESAM4 et CESAM5 ; dans ces versions antérieures, il fallait créer un module exécutable, adapté à la physique utilisée. Pour exploiter un fichier de données de CESAM4 ou de CESAM5 avec CESAM2k, il faut indiquer la physique à utiliser, et introduire les données manquantes. Pour ce faire, CESAM2k utilise les données et les options physiques explicitées dans un fichier de nom physique45 placé dans l'environement du calcul. On trouvera un exemple du fichier physique45 au § 4.11 (Page 48). En l'absence d'un tel fichier, des données et des options physiques sont définies par défaut dans la routine lit_nl_45, subordonnée de la routine lit_nl. Ces options physiques sont les suivantes :

```
NOM_CHEMIN='~/SUN_STAR_DATA/'
NOM_CTES='ctes_94'
NOM_DES='des_m'
NOM_OUTPUT='no_output'
```

```
NOM_PERTM='pertm_ext'
NOM_CONV='conv_jmj'
NOM_DIFFM='diffm_mp'
NOM_DIFFT='difft_nu'
NOM_ETAT='etat_eff'
NOM_OPA='opa_yveline'
NOM_NUC='ppcno9'
NOM_NUC_CPL='NACRE'
NOM_ATM='lim_atm'
NOM_TDETAU='hopf'
```

CESAM2k accepte aussi les fichiers de données de ses premières versions d'évaluation. Le but poursuivi dans l'exploitation des anciens fichiers de données est d'offrir la possibilité de poursuivre des calculs précédents, de ce fait, seules les fonctionnalités des anciennes versions peuvent être prises en compte.

3.16 Exploitation sous UNIX & LINUX,

Sous UNIX/LINUX, il est conseillé de créer et de laisser le module exécutable `cesam2k.out` dans le sous-directory `SOURCE`. Il est pratique de créer un sous-directory propre à chaque application dans lequel, on disposera des fichiers de données de type `mon_modele.don` propres à l'application, et les fichiers d'initialisation tels que (`m020.zams`), de réglage et de personnalisation.

Par exemple, pour calculer des modèles de ξ Hya ($3M_\odot$) de la ZAMS jusqu'à 400 My :

- Créer un sous-directory `KSI_HYA`
- Copier les fichiers `m020.zams`[7] et `mon_modele.don` du sous-directory `EXPLOIT`.
- Renommer le fichier de données `ksi_hya.don`.
- Adapter la physique et les paramètres de ce fichier au problème de ξ Hya.

Pour utilisation ultérieure, lancer le calcul d'un modèle préliminaire de ZAMS `(AGEMAX = 0.d0)` adapté aux paramètres et à la physique adoptés ; d'abord avec une précision réduite `(PRECISION='np')` en utilisant le modèle d'initialisation en ASCII, puis avec `(PRECISION='pr')` en utilisant le modèle binaire `ksi_hya_B.hom` obtenu à l'étape précédente. Copier le fichier `ksi_hya_B.hom` obtenu à l'issue de cette dernière étape dans le fichier `ksi_hya_ini_B.hom` ce dernier servant, par la suite, à initialiser le calcul de modèles utilisant des paramètres numériques ou physiques (longueur de mélange, théorie de la convection, précision, *etc...*) quelque peu différents de ceux qui ont été utilisés pour construire le fichier d'initialisation.

Dans le fichier `ksi_hya.don` indiquer l'âge à atteindre : (AGEMAX = 400.d0) et lancer l'exécution : `cesam2k.out` en l'initialisant avec `ksi_hya_ini_B.hom`.

Pour faire par la suite des modèles en changeant un ou plusieurs paramètres (âge, overshoot, *etc...*), il suffit d'adapter le fichier `ksi_hya.don` à chaque cas particulier.

[7] Utiliser des fichiers d'initialisation ASCII ou binaire de $3M_\odot$ si on en dispose.

3.17 Programmes d'exploitation

Des programmes du sous directory `EXPLOIT` permettent une exploitation des fichiers ASCII ou binaires créés par CESAM2k :

- `calib2k` : Formation du fichier de données `mon_modele.don` pour calibration d'un modèle solaire.
- `des2k_ZC` : Dessin de l'évolution des zones convectives en fonction du temps.
- `des2k_abon` : Dessin des abondances en fonction de la masse ou du rayon.
- `des2k_abontc` : Dessin de l'évolution de X, Y, Z au centre, en fonction du temps.
- `des2k_abonts` : Dessin de l'évolution de X, Y, Z en surface, en fonction du temps.
- `des2k_bin` : Dessin des variables quasi-statiques à partir d'un fichier binaire.
- `des2k_dhve` : Dessin des coefficients des équations de la diffusion du moment cinétique.
- `des2k_hr` : Dessin du diagramme HR.
- `des2k_opa` : Dessin des différences relatives entre deux tables d'opacité.
- `des2k_osc` : Dessin de 1, 2 ou 3 modèles en fonction du rayon, de la masse avec des possibilités de zoom.
- `des2k_rot` : Dessins séparés des variables de la diffusion du moment cinétique.
- `des2k_rot_ext` : Dessin du profil, en fontion du temps, de la vitesse angulaire de la couche externe en km/s ou en rad/sec.
- `fichier_vent` : Programme de contruction d'un fichier de composition chimique du vent.
- `f037_2k` : Interpolation d'un modèle en des points de masse ou de rayon fixés.

Dans chacun de ces programmes, des informations concernant leur utilisation sont données sous la forme de commentaires.

3.18 Scripts d'exploitation

Les scripts UNIX suivants (en shell `csh`) du sous-directory `SCRIPTS` facilitent l'installation et l'exploitation[8] de CESAM2k :

- `calib2k_pms` : procédure de calibration d'un modèle solaire avec PMS.
- `calib2k_zams` : procédure de calibration d'un modèle solaire initialisé sur la ZAMS homogène.
- `compile2k` : script de compilation avec optimisation.
- `compile2k-dbg` : script de compilation pour debug.
- `compile2k-dbg_list` : script de compilation pour debug pour un ensemble de routines d'une liste.
- `compile2k_list` : script de compilation avec optimisation pour un ensemble de routines d'une liste, avec création ou mise à jour d'une bibliothèque.
- `evol2k_pms` : procédure pour l'évolution à partir de la PMS homogène.
- `evol2k_zams` : procédure pour l'évolution à partir de la ZAMS homogène.
- `exe2k` : procédure d'exécution d'un programme avec optimisation

[8]Les options de compilation sont celles du compilateur LINUX LF95, il convient de les adapter à celles du compilateur utilisé.

- `exe2k-dbg` : procédure d'exécution d'un programme avec debug.
- `genere_cesam2k-dbg` : script créant la bibliothèque et l'exécutable `cesam2k-dbg.out` de debug,
- `genere_cesam2k` : script créant la bibliothèque et l'exécutable `cesam2k.out` d'exploitation,
- `lib_del` : script permettant de supprimer une routine d'une bibliothèque.
- `lib_del_repl` : script permettant de supprimer, puis de remplacer, une routine dans une bibliothèque.
- `lib_repl` : script permettant de remplacer une routine dans une bibliothèque.
- `on_voit` : script permettant de créer un fichier postscript à partir d'un fichier dvi.
- `rempl2k_mod` : script permettant de compiler, remplacer dans la bibliothèque, un ou plusieurs modules et de créer le module exécutable `cesam2k.out`.
- `rempl2k_mod-dbg` : script permettant de compiler, remplacer dans la bibliothèque, un ou plusieurs modules, et de créer le module de debug exécutable `cesam2k-dbg.out`.

Chapitre 4

Personnalisation

Bien que l'exécutable soit construit un fois pour toutes, il est possible d'adapter de façon externe un certain nombre de paramètres. Ces adaptations s'effectuent, à la demande, par l'intermédiaire de fichiers *ad hoc* à disposer dans le sous-directory où s'effectue le calcul. Ne seront concernés que les modèles calculés dans cet environnement. De façon générale, dans le directory concerné, une personnalisation peut affecter, soit un modèle particulier, soit tous les modèles calculés dans ce directory. CESAM2k recherche d'abord si une personnalisation existe pour le modèle à calculer puis, à défaut, si elle existe pour tous les modèles du directory. Des exemples de tels fichiers de personnalisation se trouvent dans le sous-directory `EXPLOIT` de la distribution. On peut ainsi :

- Avoir des commentaires d'interactivité en une langue différente du français[1].
- Utiliser une mixture initiale différente de celles implémentées.
- Modifier les rapports d'abondances d'une mixture.
- Redéfinir les rapports isotopiques.
- Créer un fichier de sortie en ASCII contenant des quantités en nombre et ordre différents de ce qui est implémenté.
- Utiliser des réglages personnalisés.
- Dessiner des boites correspondant à des objectifs à atteindre, et adapter les échelles du diagramme HR tracé "on line".
- Adapter les dimensions des cadres et le nom du device à utiliser.
- Utiliser les fichiers de données d'une version antérieure de CESAM.

Par sécurité, ainsi qu'il est explicité pour chaque cas particulier, certains des fichiers de personnalisation concernant la composition chimique ne seront pris en compte que si le paramètre `modif_chim` de la NAMELIST `nl_chim` du fichier de données est `.TRUE.` *cf.* § 3.7 (Page 27).

4.1 Langue

Une alternative pour avoir des commentaires d'interactivité en une langue différente du français :

1. Si on désire **toujours** avoir des commentaires en une langue différente du français, **avant** la compilation, dans la routine `cesam` du sous-directory

[1]Seulement en anglais dans la présente version.

SOURCE, commenter ou supprimer les instructions (vers les lignes 270-278) :

```
       INQUIRE(file='langue',exist=ok)
       IF(ok)THEN
        OPEN(unit=30,form='formatted',status='old',file='langue')
        READ(30,nl_langue) ; CLOSE(UNIT=30)
       ELSE
        WRITE(*,30)
30      FORMAT(/,'Sometimes CESAM can speak english,',/,
      1 'see aide_mem2k, chapter personnalisation ')
        langue='francais'
       ENDIF
```

et décommenter l'instruction de la ligne suivante (279) :

```
c        langue='english'
```

Les commentaires en anglais seront désormais utilisés pour toute exploitation.

2. Si on désire n'avoir qu'occasionnellement les commentaires en anglais, introduire dans l'environnement d'exploitation un fichier de nom `langue` dans lequel on codera :

```
&NL_LANGUE
langue='english'
/
```

Dans toutes les exploitations effectuées dans cet environnement les commentaires apparaitront en anglais.

Par défaut, *i.e.* en l'absence de l'une des dispositions précédentes, les commentaires seront en français.

PB Dans la version actuelle de CESAM2k, seulement une partie des principaux commentaires a été traduite en anglais.

PB **Call for man power** : il est facile de transposer dans une autre langue les algorithmes créés pour l'anglais ou encore compléter et/ou améliorer la syntaxe des ces derniers, les volontaires sont bienvenus.

4.2 Mixture

Pour utiliser une mixture initiale différente[2] de celles implémentées, coder `nom_abon='mixture'` dans le fichier de données et, dans l'environnement, placer un fichier de nom `mixture` contenant la mixture désirée en Dex. CESAM2k ***refuse*** toute mixture dont la normalisation est différente de H=12. L'exemple suivant se trouve dans le sous-directory `EXPLOIT` :

```
&NL_MIXTURE
ab(1)=12.00d0 , ab(2)=10.d0 , ab(3)=2.69d0  , ab(4)=2.15d0  , ab(5)=2.60d0
ab(6)=4.55d0  , ab(7)=3.97d0  , ab(8)=4.87d0  , ab(9)=1.56d0  , ab(10)=4.08d0
ab(11)=2.33d0 , ab(12)=3.58d0 , ab(13)=2.47d0 , ab(14)=3.55d0 , ab(15)=1.45d0
```

[2]Voir aussi § 5.2 (Page 50).

```
ab(16)=3.21d0 , ab(17)=1.5d0  , ab(18)=2.52d0 , ab(19)=1.12d0 , ab(20)=2.36d0
ab(21)=-1.17d0 , ab(22)=1.02d0 , ab(23)=0.d0   , ab(24)=1.67d0 , ab(25)=1.39d0
ab(26)=3.5d0  , ab(27)=0.92d0 , ab(28)=2.25d0
/
```

On trouve, dans la routine `abon_ini`, l'ordre et l'identification des éléments par leurs indices, en fait, celle des opacités OPAL. Dès lors, ***tous*** les modèles calculés dans ce directory, s'ils utilisent l'option `nom_abon='mixture'` , auront des abondances initiales conformes à celles du fichier `mixture`. Une autre possibilité permet de personnaliser davantage ; elle consiste à donner au fichier `mixture` le nom générique du modèle avec l'extension `.mix`, Ex : `mon_modele.mix`. Dès lors, ***seul*** le modèle `mon_modele` aura la composition chimique initiale désirée. Avec l'option `mixture`, CESAM2*k* cherche ***en priorité*** à utiliser le fichier `mon_modele.mix` ; en son absence, c'est le fichier `mixture` qui est utilisé ; le calcul n'est effectué que si l'un de ces deux fichiers existe dans l'environnement.
Cette option nécessite de coder `MODIF_CHIM=.TRUE.`, *cf.* §3.7 (Page 27).

PB `Avec l'option nom_abon='mixture', les abondances relatives des métaux devraient, en principe, correspondre à celles des opacités, au moins, faute de mieux, pouvoir être prises en charge par la routine d'opacité utilisée.`

4.3 Rapports isotopiques

Pour utiliser des rapports isotopiques différents de ceux que CESAM prend par défaut, placer dans l'environnement du calcul, un fichier de nom `mon_modele.rap_iso` dans lequel seront précisés les rapports isotopiques à utiliser. Si ***tous*** les modèles de l'environnement doivent être calculés avec des rapports modifiés, donner au fichier précédent le nom de `rap_iso`. Exemple :

```
&NL_RAP_ISO
be7sbe9=1.d-25,
be7sz=1.d-29,
c13sc12=1.10d-2,
h2sh1=3.01d-5,
he3she4=1.1d-4,
he3she4z=4.185d-4,
li6sli7=7.5d-2,
mg25smg24=0.0125,
mg26smg25=0.013,
ne22sne20=6.79d-2,
n15sn14=0.366d-2,
o17so16=0.038d-2
o18so16=0.008d-2
/
```

Les noms utilisés pour les variables suggèrent les notations des rapports isotopiques `be7sbe9` : rapport isotopique ^{7}Be/^{9}Be. `he3she4z` est le rapport isotopique ^{3}He/^{4}He sur la ZAMS lorsque le deutérium initial a été transformé en ^{3}He, `be7sz` est l'abondance, quasi nulle, du ^{7}Be initial dans Z.
Par sécurité, ce fichier ne sera pris en compte, que si le paramètre `modif_chim` de la NAMELIST `nl_chim` du fichier de données est `.TRUE.` *cf.* §3.7 (Page 27).

4.4 Personnalisation des rapports d'abondances

La mixture une fois définie, par exemple `solaire_gn` (Grevesse & Noëls 93), il est possible d'y modifier les abondances. Pour ce faire, créer dans l'environnement du calcul, un fichier de nom `mon_modele.modif_mix`, dans lequel seront indiquées les modifications en DeX à apporter. Si tous les modèles de l'environnement doivent être calculés avec ces modifications, donner au fichier précédent le nom de `modif_mix`. Exemple se trouvant dans le directory EXPLOIT :

```
&nl_modif_mix
add_Li=2.d0 ; add_Be=0.d0 ; add_B=0.d0
add_C=-1.0d0 ; add_N=0.d0 ; add_O=-0.3d0 ; add_F=0.d0 ; add_Ne=0.d0
add_Na=0.d0 ; add_Mg=0.d0 ; add_Al=0.d0 ; add_Si=0.d0 ; add_P=0.d0
add_S=0.d0 ; add_Cl=0.d0 ; add_Ar=0.d0 ; add_K=0.d0 ; add_Ca=0.d0
add_Sc=0.d0 ; add_Ti=0.d0 ; add_V=0.d0 ; add_Cr=0.d0 ; add_Mn=0.d0
add_Fe=0.d0 ; add_Co=0.d0 ; add_Ni=0.d0 ; add_Z=-1.d0
/
```

Les notations suggèrent les quantités à modifier ; ainsi `add_Li=2.d0` est le nombre de DeX à ajouter à l'abondance du lithium. `add_Z=-1.d0` signifie que les abondances ***de tous les métaux*** seront diminuées d'un DeX. ***Par sécurité***, ce fichier ne sera pris en compte que si le paramètre `modif_chim` de la NAMELIST `nl_chim` du fichier de données est `.TRUE.` *cf.* § 3.3 (Page 22).

Si on a codé `GARDE_XISH=.FALSE.` dans le fichier de données *cf.* § 3.7.1 (Page 28), le paramètre `add_Z`, qui agit sur les métaux dans leur ensemble, n'a pas d'incidence dans la détermination des rapports métal/Z. A l'opposé, une valeur non nulle de `add_C`, par exemple, aura une incidence sur les rapports métal/Z.

4.5 Personnalisation de la composition chimique du vent.

Dans le cas standard, avec la donnée `mdot` non nulle, *cf.* § 3.3 (Page 22), la composition chimique de la masse perdue ou gagnée par le vent est celle de la couche la plus externe du modèle. Il est possible de personnaliser cette composition chimique, afin qu'elle diffère de celle de la couche externe. Pour ce faire, placer dans l'environnement du calcul, un fichier de nom `mon_modele.vent` dans lequel seront indiquées les fractions de masse de chaque élément chimique dans l'apport ou la perte de masse. Si tous les modèles de l'environnement doivent être calculés avec ces modifications, donner au fichier précédent le nom de `vent`. Un exemple se trouve dans le sous-directory EXPLOIT :

```
&nl_vent
vt_H=0.7347,vt_He=0.2483,vt_Li=1.033e-08,vt_Be=1.72816e-10,
vt_B=4.859e-09,vt_C=2.899e-03,vt_N=8.493e-04,vt_O=7.885e-03,
vt_F=4.182e-07,vt_Ne=1.768e-03,vt_Na=3.501e-05,vt_Mg=6.736e-04,
vt_Al=6.078e-05,vt_Si=7.434e-04,vt_P=8.198e-06,vt_S=3.704e-04,
vt_Cl=4.924e-06,vt_Ar=7.315e-05,vt_K=3.845e-06,vt_Ca=6.541e-05,
vt_Sc=4.126e-08,vt_Ti=3.041e-06,vt_V=3.888e-07,vt_Cr=1.856e-05,
vt_Mn=1.357e-05,vt_Fe=1.287e-03,vt_Co=3.492e-06,vt_Ni=7.6092e-05
/
```

Les notations sont suggestives, ainsi `vt_Li=1.033e-08` est la fraction masse du lithium dans le vent. Une renormalisation étant effectuée lors de l'utilisation, il n'est pas nécessaire de normaliser les fractions de masse figurant dans ce fichier, certaines peuvent être nulles et même négatives. Suivant le signe de `MDOT`, la quantité de matière, apportée ou soustraite, est supposée provenir de la zone convective externe qui existe toujours. Le fichier `vent` peut être construit[3] par le programme `fichier_vent` du sous-directory EXPLOIT.

Par sécurité, ce fichier ne sera pris en compte, que si le paramètre `modif_chim` de la NAMELIST `nl_fdon` du fichier de données est `.TRUE.` *cf.* § 3.7 (Page 27).

4.6 Chute de planètoïdes

Au cours d'une évolution CESAM2*k* permet de simuler une chute de planètoïdes sur un intervalle de temps limité. Pour ce faire, placer dans l'environnement du calcul, un fichier de nom `mon_modele.planet` dans lequel seront indiquées les fractions de masse de chaque élément de la composition chimique des planètoïdes, le nombre total de planètoïdes qui sera reçu par l'étoile, l'intervalle de temps concerné par la chute et le profil de l'intensité de la chute. Si tous les modèles de l'environnement doivent être calculés avec ces modifications, donner au fichier précédent le nom de `vent`. Un exemple se trouve dans le sous-directory EXPLOIT :

```
&nl_planet
vt_H=0.7347,vt_He=0.2483,vt_Li=1.033e-08,vt_Be=1.72816e-10,
vt_B=4.859e-09,vt_C=2.899e-03,vt_N=8.493e-04,vt_O=7.885e-03,
vt_F=4.182e-07,vt_Ne=1.768e-03,vt_Na=3.501e-05,vt_Mg=6.736e-04,
vt_Al=6.078e-05,vt_Si=7.434e-04,vt_P=8.198e-06,vt_S=3.704e-04,
vt_Cl=4.924e-06,vt_Ar=7.315e-05,vt_K=3.845e-06,vt_Ca=6.541e-05,
vt_Sc=4.126e-08,vt_Ti=3.041e-06,vt_V=3.888e-07,vt_Cr=1.856e-05,
vt_Mn=1.357e-05,vt_Fe=1.287e-03,vt_Co=3.492e-06,vt_Ni=7.6092e-05,
ypl=0.d0,zpl=0.9d0,
n_planet=10,profil='rectangle',age_deb=5.d0,age_fin=10.d0,
r_giration=1.d0,t_giration=1.d0
/
```

Les notations sont suggestives. Avec `vt_H` et `vt_He` ***non nuls***, des valeurs non nulles de `ypl` et/ou `zpl` permettent de modifier les fractions de masse de l'hydrogène `X`, de l'hélium `Y` et des métaux `Z` des planètoïdes. Avec le fichier `planet` précédent, l'abondance en masse de l'hélium sera $Y = 0$, celle des métaux $Z = 0.9$ et celle d'hydrogène $X = 1.0 - 0.0 - 0.9 = 0.1$. `age_deb` et `age_fin` sont respectivement les âges, en million d'années, du dédut et de la fin de la chute. `r_giration` est le rayon de giration en UA et `t_giration` le temps de giration, en années. CESAM2*k* estime la vitesse angulaire des planètoïdes en admettant qu'ils décrivent un arc de $\pi/2$ pendant un temps de giration. Une valeur négative du temps de giration, correspond à un apport négatif de moment cinétique, i.e. rétrograde. Avec une valeur nulle du temps de giration il n'y a pas d'apport de moment cinétique.
Dans la routine `planetoides.f`, CESAM2*k* propose 4 profils :

1. `rectangle` : L'intensité de la chute est constant.
2. `triangle` : Le profil de l'intensité de la chute est triangulaire, le maximum étant centré au milieu de l'intervalle de temps.

[3]La composition chimique de l'exemple est celle de la mixture météoritique de Grevesse & Sauval 98. Par ailleurs, il n'est pas nécessaire, ici, de respecter la 'case'.

3. `parabole` : Le profil de l'intensité de la chute est parabolique, le maximum étant centré au milieu de l'intervalle de temps.
4. `gauss` : Le profil de l'intensité de la chute est une gaussienne limitée à l'intervalle de temps. Le maximum est centré au milieu de l'intervalle de temps. L'écart type est arbitrairement pris égal au tiers de l'intervalle de temps.

Cette option nécessite de coder `MODIF_CHIM=.TRUE.`, *cf.* § 3.7 (Page 27).

4.7 Ascii

Pour créer un fichier de sortie en ASCII personnalisé, coder `nom_output='ascii'` dans le fichier de données et, dans l'environnement, placer un fichier de nom `sortie_ascii`[4] dans lequel sont définis les paramètres créant la structure souhaitée. Le codage, ***délicat***, est décrit dans la routine `output.f`, les indices des variables sont identifiés dans la routine `ascii.f` du sous-directory `SOURCE`. Dans l'exemple :

```
12  3
1 2 3 4 5 6 9 10 11 12 13 15
1 2 3
.FALSE.
.TRUE.
.TRUE.
Fichier pour sortie ascii:
```

les quantités ont les significations suivantes :
- 12 3 ⟶ on écrira 12 variables globales et 3 variables locales,
- 1 2 3 4 5 6 9 10 11 12 13 15 ⟶ indices des 12 variables globales (mstar*msol, rtot*rsol, ltot*lsol, z0, x0, alpha, X dans ZC, Y dans ZC, d2p, d2ro, age, vsal, w_rot initial).
- 1 2 3 ⟶ indices des 3 variables locales : Rayon, log(M/Mtot), Température).
- .FALSE. ⟶ la masse sera écrite en fraction de la masse totale et non en DeX.
- .TRUE. ⟶ tabulation du centre à la surface.
- .TRUE. ⟶ on ajoute la composition chimique locale à la suite des variables locales.
- Fichier pour sortie ascii : texte d'entête du fichier de sortie.

Le nom générique du fichier de données est repris pour former celui du fichier de sortie en ASCII : avec `mon_modele.don` le nom du fichier de sortie est `mon_modele-ascii`. Avec le fichier cité comme exemple, on a obtenu :

```
Fichier pour sortie ascii: mon_modele-ascii
CESAM2k version 0.0.0.0 lagr colloc  1  2 np no diffus, 06 Juillet 2003 17h46
Physique utilisée: etat_eff, opa_yveline_lisse, conv_jmj, ppcno9, NACRE
solaire_gn, lim_atm, hopf, perte_ext, diffm_mp, difft_nu, ctes_94
 10  H1  He3  He4  C12  C13  N14  N15  O16  O17  Si28
      541        12        13        10        -1
 1.989190000000E+33 6.010741880620E+10 2.478601588572E+33 2.000000000000E-02 7.000000000000E-01
 1.800000000000E+00 7.000000000000E-01 2.800000000000E-01-4.241618755712E+01-2.519592625187E+01
 0.000000000000E+00 0.000000000000E+00
 0.000000000000E+00 0.000000000000E+00 1.359013041915E+07 7.000000000000E-01 8.826927693868E-05
 2.799117307231E-01 3.425410508692E-03 4.128413464627E-05 1.059170894932E-03 4.168107978225E-06
 9.641655715570E-03 3.903146720978E-06 5.824407491460E-03
```

[4]Dont un exemple se trouve dans le sous-directory `EXPLOIT`.

```
1.302042603905E+09 1.175206179724E-08 1.353639070045E+07 7.000000000000E-01 8.826927693868E-05
2.799117307231E-01 3.425410508692E-03 4.128413464627E-05 1.059170894932E-03 4.168107978225E-06
9.641655715570E-03 3.903146720978E-06 5.824407491460E-03
        ..................................................

6.014328252162E+10 1.000000000164E+00 4.543950048106E+03 7.000000000000E-01 8.826927693868E-05
2.799117307231E-01 3.425410508692E-03 4.128413464627E-05 1.059170894932E-03 4.168107978225E-06
9.641655715570E-03 3.903146720978E-06 5.824407491460E-03
6.014636429604E+10 0.000000000000E+00 4.543482782113E+03 7.000000000000E-01 8.826927693868E-05
2.799117307231E-01 3.425410508692E-03 4.128413464627E-05 1.059170894932E-03 4.168107978225E-06
9.641655715570E-03 3.903146720978E-06 5.824407491460E-03
```

Comme pour les abondances initiales *cf.* § 4.2 (Page 38), il est possible de ne créer ce fichier de sortie personnalisé, que pour un modèle particulier, en donnant au fichier de sortie `sortie_ascii` le nom du modèle avec l'extension `.ascii`, exemple : `mon_modele.ascii`. S'il existe dans l'environnement, ce seront les paramètres de ce dernier fichier qui seront pris en compte pour ***le*** modèle de nom `mon_modele`. En son absence, CESAM2*k* cherchera à utiliser ceux du fichier `sortie_ascii`, à défaut de ce dernier, il n'y aura pas de sortie ASCII.

4.8 Réglages

Les réglages, définis dans la routine `cesam.f` en fonction du paramètre `PRECISION` du fichier de données `mon_modele.don`, peuvent être personnalisés en codant : `PRECISION='rg'`, et en disposant dans l'environnement un fichier de nom : `reglages`[5] comportant les réglages à utiliser. Exemple d'un tel fichier `reglages` :

```
&NL_RLG
m_qs=2,
m_ch=3,
m_rot=2,
m_tds=2,
m_ptm=2,
ordre=1,
precix=1.d-3,
precit=0.15d0,
ro_test=0.1d0,
psi0=0.05d0,
d_grav=0.5d0,
loc_zc=1.d-3,
dtmax=200.d0,
dt0=1.d0,
ini0=5,
n_atm=75,
kipp=.FALSE.,
en_masse=.TRUE.,
ctel=0.0d0,
ctep=-1.d0,
```

[5]Dont un exemple se trouve dans le sous-directory `EXPLOIT`.

```
ctem=15.d0,
cter=0.0d0,
ctet=0.d0,
mvt_dis=.FALSE.,
dn_fixe=0.05d0,
dpsi=0.05d0,
w_form=.05d0,
nom_coeff_rota='coeff_rota_saha',
ecrit_rot=.TRUE.
/
```

Les significations des paramètres sont les suivantes :

m_qs=2 : Ordre des B-splines pour les variables quasi-statiques.

m_ch=3 : Ordre des B-splines pour l'interpolation de la composition chimique.

m_rot=3 : Ordre des B-splines pour l'interpolation du moment cinétique.

m_tds=2 : Ordre des B-splines pour interpolation de l'énergie graviphique.

m_ptm=2 : Ordre des B-splines pour interpolation de la perte de masse.

ordre=1 : Ordre du schéma d'intégration des réactions nucléaires avec `rk_imps`.

precix=1.d-4 : Précision sur les itérations Newton-Raphson des intégrations spatiales.

precit=0.1d0 : Variation relative maximale pour l'intégration temporelle de la composition chimique.

ro_test=0.1d0 : Test de variation de l'énergie graviphique si `ro > ro_test`.

psi0=0.05d0 : Constante de répartition à assurer.

d_grav=0.5d0 : Variation temporelle maximale de l'énergie graviphique.

loc_zc=5.d-4 : Précision de la localisation des limites zones radiatives / zones convectives.

dtmax=200.d0 : Pas temporel maximum.

ini0=5 : Nombre d'itérations Newton-Raphson avec réestimation de la composition chimique, de la vitesse angulaire et des limites zones radiatives / zones convectives.

n_atm=75 : Nombre de couches dans l'atmosphère restituée.

kipp=.TRUE. : Utilisation de l'approximation de Kippenhahn.

en_masse=.TRUE. : Variables quasi-statiques lagrangiennes.

ctel, ctep, ctem, cter, ctet : Facteurs de répartition.

mvt_dis=.TRUE. : Ajustement de la composition chimique due aux déplacements des discontinuités..

dn_fixe=0.05d0 : Taux limite de variation du nombre de couches nécessitant un réajustement de la grille fixe pour l'interpolation de la composition chimique.

dpsi=0.02 : Taux limite de variation de la fonction de répartition nécessitant une modification du nombre de couches.

w_form=0.05 : Coefficient, de valeur absolue inférieure à l'unité, permettant de modifier le profil de la vitesse angulaire initiale.

nom_coeff_rota='coeff_rota_saha' : Nom de la routine de calcul des coefficients des équations de la diffusion du moment cinétique.

ecrit_rot=.TRUE. : Ecriture, dans l'environnement du fichier `mon_modele_coeff_rota.dat` permettant le dessin des coefficients de la diffusion du moment cinétique.

L'ajustement des constantes de répartition ***est délicat***. Les valeurs par défaut sont `ctel=0, ctep=-1, ctem=15, cter=0, ctet=0`. Prendre garde que `ctep` et `ctet` doivent être ***négatives***, car elles affectent respectivement la pression et la température, quantités décroissantes du centre à la surface.

Comme pour les abondances initiales *cf.* § 4.2 (Page 38), il est possible de plus personnaliser ces réglages en donnant au fichier des réglages, celui du modèle avec l'extension `.rg`, exemple : `mon_modele.rg`. S'il existe dans l'environnement, ce seront les paramètres de ce fichier qui seront pris en compte pour ***le*** modèle de nom `mon_modele`. En son absence, ce seront les quantités du fichier `reglages`, s'il n'existe pas, le calcul ne sera pas effectué.

4.9 Zoom

Il est possible de personnaliser le dessin "on line" du diagramme HR en y dessinant un rectangle cible à atteindre par le trajet évolutif, ainsi que de modifier l'échelle du diagramme, en centrant celui-ci autour de la valeur du point cible à des distances précisées en DeX de $\log T_{\rm eff}$ et $\log L$. On peut aussi tracer une cible de [Fe/H] à atteindre, dans le cadre du dessin des abondances en fonction de la masse. Pour ce faire, coder `nom_des='zoom'` dans le fichier de données et, dans l'environnement, placer un fichier de nom `zoom`, dans lequel sont introduites les données nécessaires. Exemple d'un tel fichier `zoom`[6] :

```
&NL_DES
teff_des=6530.,
dteff_des=50., -30.,
zoom_t=0.2, -0.1,
l_des=6.844,
dl_des=0.6, -0.4
zoom_l=0.1, -0.2,
fesh_des=0.05,
dfesh_des=0.03, -0.04,
logteff_max=4.6,
logteff_min=3.5,
logl_max=5.,
logl_min=2.5
/
```

– Il n'y a pas de personnalisation si `teff_des` et/ou `l_des` est négatif ou nul.

[6]Exemple se trouvant dans le sous-directory `EXPLOIT`.

- Le zoom n'est effectif qu'avec `zoom_t` et `zoom_l` non nuls.
- Pour ne pas dessiner de cible en métallicité, coder `fesh_des=100`.
- Les valeurs des abscisses et ordonnées maximales *i.e.* `logteff_max (min), logl_max (min)` ***prévalent*** sur celles des cibles ; dans l'exemple précédent, les échelles logarithmiques en température effective et en luminosité variront respectivement dans les intervalles $[4.6, 3.5]$ et $[2.5, 5.]$.
- Pour **seulement** adapter les échelles du diagramme HR coder, par exemple :

```
&NL_DES
teff_des=0.,
dteff_des=0., 0.,
zoom_t=0., 0.,
l_des=0.,
dl_des=0., 0.
zoom_l=0., 0.,
fesh_des=0.,
dfesh_des=0., 0.,
logteff_max=3.6,
logteff_min=2.5,
logl_max=2.,
logl_min=-2.5
/
```

- Pour **seulement** cadrer le diagramme HR autour de la cible, coder :

```
&NL_DES
teff_des=6530.,
dteff_des=50., -30.,
zoom_t=0.2, -0.1,
l_des=6.844,
dl_des=0.6, -0.4
zoom_l=0.1, -0.2,
fesh_des=100.,
dfesh_des=0., 0.,
logteff_max=0.,
logteff_min=0.,
logl_max=0.,
logl_min=0.
/
```

La signification des variables est la suivante :

teff_des=6530. : Température effective du centre de la cible.

dteff_des=50., -30. : Ecarts en température effective.

zoom_t=0.2, -0.1 : Extension en température effective autour de la cible en DeX.

l_des=6.844 : Luminosité du centre de la cible.

dl_des=0.6, -0.4 : Ecarts en luminosité.

zoom_l=0.1, -0.2 : Extension en luminosité autour de la cible en DeX.

fesh_des=0.05 : Cible à atteindre en métallicité.

dfesh_des=0.03, -0.04 : Ecarts de la cible en métallicité.

logteff_max=4.6 : $\log_{10}$ de l'abscisse maximale du diagramme HR.

logteff_min=3.5 : $\log_{10}$ de l'abscisse minimale du diagramme HR.

logl_max=5. : $\log_{10}$ de l'ordonnée maximale du diagramme HR.

logl_min=2.5 : $\log_{10}$ de l'ordonnée minimale du diagramme HR.

Il est possible de personnaliser davantage, en donnant au fichier `zoom` le nom attaché au modèle avec l'extension `.zoom`, exemple : `mon_modele.zoom`. Dès lors, les dispositions précédentes ne seront effectives ***que pour*** le modèle `mon_modele`. Avec l'option `zoom`, CESAM cherche à utiliser le fichier `mon_modele.zoom`, en son absence, le fichier `zoom` est utilisé.

4.10 Device

Pour adapter les dimensions des cadres et le nom du device à utiliser, placer dans l'environnement du calcul, un fichier de nom `device`, dans lequel sont introduites les données nécessaires. Exemple d'un tel fichier `device`[7] :

```
&NL_DEVICE
h=6.5,
dh=1.5,
ld=8.5,
dl=2.,
xleft=1.8,
ybot=1.4,
device='/xw'
/
```

La signification des variables est la suivante (unités en cm) :

h=6.5 : Hauteur des cadres.

dh=1.5 : Espace vertical entre les cadres.

ld=8.5 : Largeur des cadres.

dl=2. : Espace horizontal entre les cadres.

xleft=1.8 : Distance du dessin à la limite gauche de l'écran.

ybot=1.4 : Distance du dessin à la limite inférieure de l'écran.

device='/xw' : Nom du device.

On peut aussi modifier directement les valeurs de ces paramètres dans le module `mod_donnees` du sous-directory `SOURCE`.

[7] Exemple se trouvant dans le sous-directory `EXPLOIT`.

4.11 Physique45

Un fichier, de nom `physique45`, placé dans l'environnement du calcul, *cf.* § 3.15 (Page 33), permet d'utiliser un fichier de données de CESAM4 ou de CESAM5 avec une physique différente de celle prévue par défaut. Par exemple :

```
&NL_PHYSIQUE45
 NOM_CHEMIN='~/SUN_DATA/'
 NOM_CTES='ctes_85'
 NOM_DES='des_r'
 NOM_OUTPUT='osc_adia'
 NOM_PERTM='pertm_ext'
 NOM_CONV='conv_cm'
 NOM_DIFFM='diffm_0'
 NOM_DIFFT='difft_nu'
 NOM_ETAT='etat_ceff'
 NOM_OPA='opa_yveline'
 NOM_NUC='ppcno3ac10'
 NOM_NUC_CPL='Cau-Fow'
 NOM_ATM='lim_atm'
 NOM_TDETAU='roger_00'
/
```

On trouvera ce fichier dans le sous-directory `EXPLOIT`.

Chapitre 5

Extensions

5.1 Ajout d'une chaîne de réactions thermonucléaires

Pour implanter une nouvelle chaîne de réactions thermonucléaires, il est souvent nécessaire d'ajouter un ou plusieurs isotopes, et aussi une ou plusieurs réactions thermonucléaires.

– Pour ajouter un isotope[1] :

1. Dans le module `mod_nuc` augmenter le `PARAMETER niso_tot` d'une unité, Exemple : `niso_tot=28`.
2. Dans la routine `taux_nuc` :
   (a) Dans les déclarations, ajouter l'excès de masse de l'isotope et son symbole, Exemple : `Fe56=-60.6054d0`.
   (b) Ajouter la masse de l'isotope, (il est conseillé d'ajouter la masse de l'isotope, dans les routines de constantes physiques `ctes_85` et `ctes_95`, et par suite dans les déclarations du module `mod_donnees` et dans l'instruction de restriction `ONLY` de la routine `taux_nuc`, Exemple : `nucleot(27)=afe56, afe56=55.847d0, afe56, af18...`).
   (c) Ajouter la charge de l'isotope, Exemple : `zit(28)=11`.
   (d) Ajouter le nom de l'isotope, Exemple : `'B11 '`. (ces chaînes de 4 caractères sont à cadrer à gauche sauf pour les symboles de 2 caractères qui sont centrés.).
   (e) Si besoin, ajouter un rapport isotopique pour ce nouvel isotope :
      i. Introduire sa valeur dans la liste disposée à la fin de la routine.
      ii. Indiquer son nom dans la NAMELIST `nl_rap_iso` dans la partie déclarative de la routine.
      iii. Ajouter son nom dans la partie déclarative du module `mod_nuc`.

[1] CESAM2k prévoit la possibilité d'utiliser des éléments chimiques de l'hydrogène ($Z = 1$) jusqu'au nickel ($Z = 28$). Pour introduire des éléments de masse supérieure, les disposer dans la routine `abon_ini` en appliquant une procédure voisine de celle décrite pour l'ajout d'un isotope.

– Pour ajouter une réaction :
  1. Dans la routine `taux_nuc` :
     (a) Compléter la liste de commentaires en y indiquant le symbole de la réaction et son indice, Exemple : `réaction 46 : B11(p,g)C12`.
     (b) Initialiser le nom de la réaction, le défaut de masse, et les charges des noyaux concernés, Exemple : `nom_react(2)='H2(p,g)He3' nuc=H2+p-He3`; `qt(2)=nuc`; `izzt(2,1)=1`; `izzt(2,2)=1`.
     (c) Entrer le taux de la réaction[2] `rt(i)` en ln et ne pas oublier d'y inclure les 1 ! ou 2 ! ou 3 ! du dénominateur, Exemple : `réaction 8 : C12(p,g)N13(e+ nu)C13 z0=6, z1=1........` en recalculant, au besoin, les coefficients à partir des $S(0)$, $S'(0)$, $S"(0)$ suivant l'équation 4-49 de Lang Astrophysical formulae [1968].
     (d) Créer la routine de réactions thermunucléaires en s'inspirant d'une des routines existantes, `ppcno12BeBFe.f` par exemple.
  2. Dans la routine `tabul_nuc`
     (a) Entrer les paramètres du calcul des taux, le nombre de réactions et les correspondances des indices, Exemple : `CASE('ppcno12') ; WRITE(2,6) ; WRITE(*,6)....`
  3. Dans le module `mod_nuc` :
     (a) Introduire par un `INCLUDE` le nom de la routine de la chaîne de réactions, Exemple : `INCLUDE 'ppcno9.f'`
  4. Dans la routine générique `nuc` :
     (a) Rendre active et faire reconnaître la nouvelle routine.
  5. Tester les algorithmes ainsi modifiés en exploitant le programme `test_jacobien_reac_nuc.f` du sous-directory TEST.

5.2 Ajout d'une mixture initiale

Ainsi qu'il est décrit au § 4.2 (Page 38), il est possible de personnaliser les abondances initiales à l'aide d'un fichier `mixture`. Quand la mixture initiale est d'intérêt général[3] il est préférable et très simple, d'en ajouter la table des abondances dans la routine `abon_ini` de la source, en lui donnant un nom d'au plus 20 caractères. Pour refaire ensuite le module exécutable, il suffit d'appliquer le script `rempl2k_mod` au module `mod_nuc` : `rempl2k_mod mod_nuc.`

[2] C'est pour des raisons historiques que ces calculs sont effectués en ln. Les raisons étaient d'éviter d'une part, les erreurs de troncature et d'autre part, les dépassements de capacité avec la limitation à 10^{38}.

[3] Pour faire profiter la communauté de votre travail, communiquer la routine ainsi transformée aux responsables de la mise à jour de CESAM2*k*.

Chapitre 6

Debug

6.1 Debug

La structure de CESAM2k en modules permet une vérification efficace de la syntaxe par le compilateur. Le revers de la médaille est que la mise au point et/ou le debug nécessitent de recompiler tout le module auquel appartient la routine dans laquelle le bug est recherché. Lors de l'installation, il est utile de créer, conjointement à la bibliothèque d'exploitation `libcesam2k.a` une bibliothèque de debug `libcesam2k-dbg.a`. Cette bibliothèque est créée automatiquement lors de l'exécution du script `genere_cesam2k-dbg` du sous-directory `SCRIPTS`. Pour ce faire, utiliser les options de debug du compilateur ; la procédure `compile2k-dbg` du sous-directory `SCRIPTS` en est un exemple. Le temps nécessaire à la construction de la bibliothèque de debug est significativement plus court que celui de la construction de la bibliothèque d'exploitation, facteur 5 à 10 par exemple, facteur qui se retrouve, mais dans l'autre sens, lors d'une exécution.

Exemple : un bug a été détecté dans la routine `des_m` du module `mod_cesam`[1]. On introduit des instructions de debug dans `des_m`. Dans le sous-directory `TESTS` on exécute le programme `test_cesam` dans lequel on inclut le module `mod_cesam` :

```
INCLUDE '../SOURCE/mod_cesam.f'

c******************************************************************

PROGRAM test_cesam

USE mod_cesam

IMPLICIT NONE

CALL cesam

STOP
```

[1]Les *bons* compilateurs indiquent le nom des routines où ils détectent un bug.

```
END PROGRAM test_cesam
```

Au link, le fichier `mod_cesam.mod` créé dans le sous-directory `TESTS` sera utilisé et non pas celui du sous-directory `SOURCE`. Lors de l'exécution, les instructions de debug, alors prises en compte, permettront de détecter le défaut. Une fois la routine `des_m` corrigée, dans le sous-directory `SOURCE`, l'exécution du script : `mod_repl mod_cesam`, remplacera le module `mod_cesam` dans la bibliothèque et recréera l'exécutable `cesam2k.out`. Ainsi, aucun fichier du sous-directory `SOURCE` n'a été déplacé.

6.2 Bugs connus

1. Erreurs à la compilation.
   (a) Dans des routines de provenance externe *e.g.* `etat_ceff, etat_eff, ferdir, ..` les variables ne sont pas déclarées, ou encore sont déclarées de façon implicite. Il y a risque d'erreur ou de “warning” lors de la compilation. ***Remède*** : supprimer l'option `IMPLICIT NONE`.
   (b) Avec le compilateur F95 de DIGITAL, on obtient un 'compilor internal error' lors de la compilation du module `mod_etat`. La routine `etat_eff` est en cause. ***Remède*** : utiliser un degré d'optimisation au plus égal à 2.
   (c) Avec un compilateur dont la version n'est pas à jour de la norme F95, (comme certaines versions du compilateur `ifc`) il y a refus d'effectuer une allocation dans une routine autre que celle où est déclarée l'allocation. ***Remède*** : remplacer dans les routines concernées, les instructions `ALLOCATABLE` par `POINTER` pour les tableaux incriminés.
2. Mauvaise convergence, le processus itératif ”tourne en rond” : la convergence s'amorce, puis les corrections ne diminuent plus, ou oscillent autour d'une valeur fixe. ***Cause possible*** : interpolation non dérivable de l'opacité *e.g.* avec des opacités interpolées linéairement. ***Remède*** : si l'augmentation de l'ordre d'interpolation des opacités est la solution théorique, elle est souvent impraticable, et il y a risque d'oscillations à cause de pas d'interpolation trop grands, ou de raccords pas assez lisses entre tables ; pratiquement, il faut se contenter d'une précision moins grande pour passer le point difficile. Ce type de difficulté se produit assez souvent dans l'atmosphère, à cause du raccord entre tables d'opacités d'origines différentes. Dans certaines conditions, CESAM2k *”force la convergence”*.
3. Mauvais fonctionnement du contrôle de la variation temporelle de l'entropie spécifique (TdS), on obtient le message : ”TdS varie trop... diminution du pas temporel”, le pas temporel diminue sans que, pour autant, le problème disparaisse. ***Cause possible*** : discontinuité temporelle de la composition chimique avec un cœur convectif. ***Remède*** : parfois, bien que celà semble paradoxal et que l'on n'en comprenne pas la raison, il suffit de reprendre l'évolution avec un pas temporel plus grand pour que le point délicat soit passé ; on peut aussi tenter de modifier `d_grav`, valeur du paramètre de contrôle de la variation

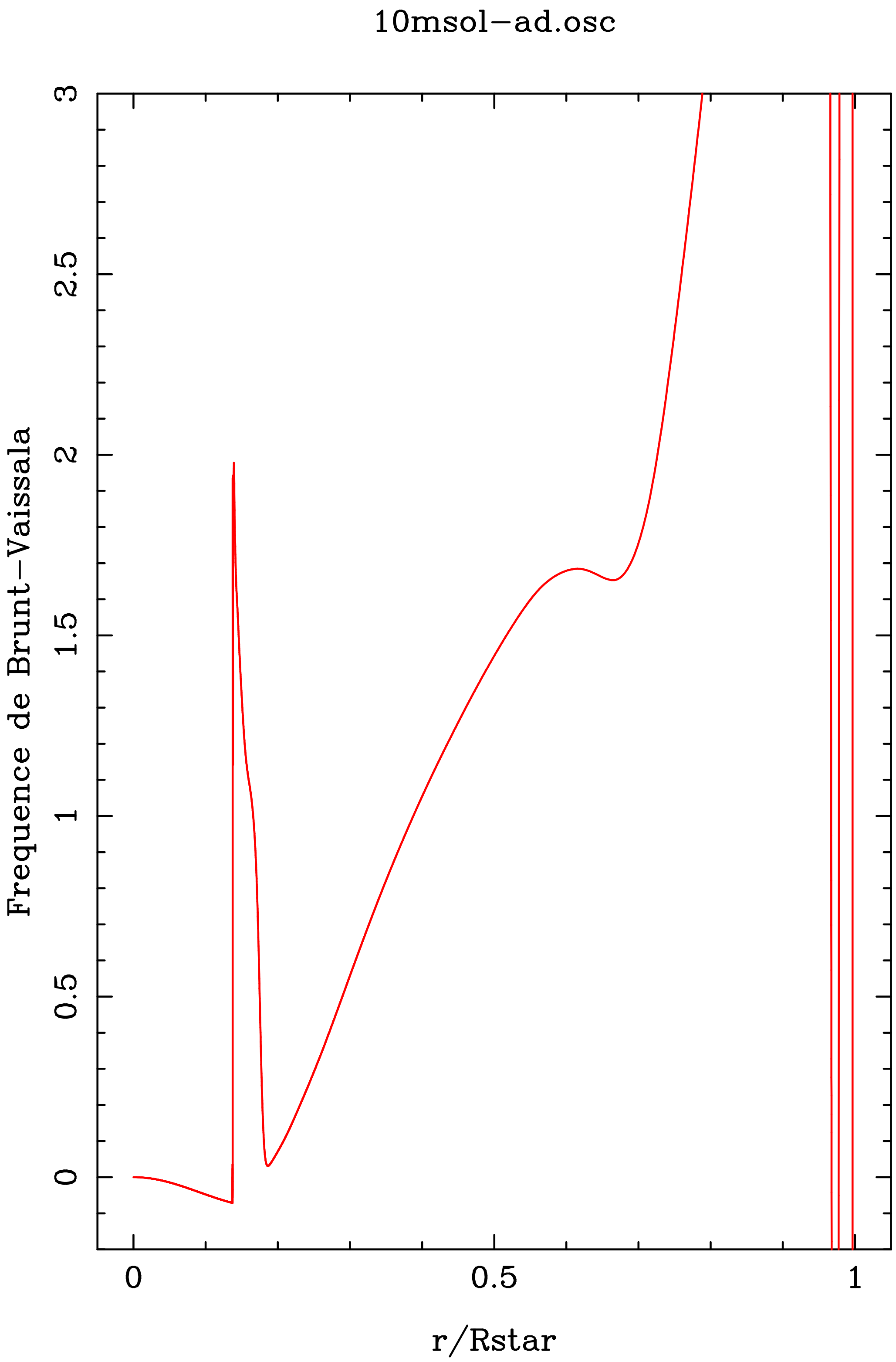

FIG. 6.1 – Profil de la fréquence de Brunt-Väissälä d'un modèle de $10M_\odot$ de séquence principale ($X_c = 0.35$), calculé avec la prodédure décrite § 16 (Page 55).

admise du TdS ; augmenter sa valeur, diminue la précision mais augmente la robustesse ; on peut éliminer le contrôle en donnant une valeur très grande (10^{10} à ce paramètre de contrôle ; il est initialisé dans la routine `cesam`, *cf.* § 8.10 (Page 147). On contourne parfois la difficulté en utilisant l'approximation de Kippenhahn *cf.* § 7.4.1 (Page 83). Pour ajuster ces paramètres il est pratique d'utiliser la possibilité de personnaliser les réglages de CESAM2*k* *cf.* § 4 (Page 37).

4. Pour une évolution avec PMS, en précision normale `np`, une fois le modèle initial obtenu, il n'y a pas convergence pour le premier pas temporel. ***Cause possible*** : La cause de cette difficulté mineure, n'a pas été trouvée. ***Remède*** : CESAM2*k* divise le pas temporel par 2 et la convergence est obtenue sans difficulté.
5. Divergence au premier pas temporel. ***Cause possible*** : pas temporel trop élevé. ***Remède*** : diminuer le `dt0` initialisé dans la routine `cesam` suivant le type de précision requis. Pour ajuster ces paramètres il est pratique d'utiliser la possibilité de personnaliser les réglages de CESAM2*k* *cf.* § 4 (Page 37).
6. La localisation des limites zones radiatives / zones convectives sur un point de grille ne fonctionne pas. ***Cause possible*** : les gradients $\nabla_{\rm rad}$ et $\nabla_{\rm ad}$ ne se croisent pas franchement, ou encore la limite se déplace notablement *i.e.* , lors de la disparition d'un cœur convectif. ***Remède*** : quand les gradients sont voisins, la limite zone radiative / zone convective est mal définie, y vouloir forcer un point de grille exactement est illusoire et inutile, CESAM2*k* s'accomode de la situation en imposant la position obtenue après le nombre d'itérations fixé dans la routine `cesam` suivant le type de précision requis. Pour ajuster ces paramètres, il est pratique d'utiliser la possibilité de personnaliser les réglages de CESAM2*k* *cf.* § 4 (Page 37).
7. Impossibilité de construire un modèle de séquence principale d'âge zéro. ***Cause possible*** : utilisation d'une composition chimique trop éloignée de l'équilibre *e.g.* celle utilisée pour les modèles de pré-séquence principale avec les abondances cosmiques de H2 et Li7. ***Remède*** : adapter la composition chimique initiale. Pour ajuster ces paramètres, il est pratique d'utiliser la possibilité de personnaliser les abondances initiales de CESAM2*k*, *cf.* § 4 (Page 37).
8. Pour des masses élevées $\sim 10M_\odot$ et plus, après la disparition de l'hydrogène central, des ZC apparaissent, disparaissent nuisant à la convergence. ***Cause possible*** : présence de zones semi-convectives. ***Remède*** : difficulté chronique, CESAM2*k* ne prévoit pas de traitement de la semi-convection.
9. Convergence impossible en utilisant le terme correctif de pression turbulente $\frac{d\ln P_{\rm gaz}}{d\ln P}$ *i.e.* avec une valeur strictement négative pour l'argument `cpturb` de la NAMELIST `NL_CONV`. ***Cause possible*** : Ce problème résulte vraisemblablement d'une incohérence de la MLT. ***Remède*** : néant en l'état actuel.
10. Le programme `lit_osc` du sous-directory `EXPLOIT` de la source refuse de lire un fichier d'extension `.osc`. ***Cause possible*** : Pour une raison inconnue, la première ligne du fichier `mon_modele.osc` est "repliée" et le fichier est alors illisible. ***Remède*** : "déplier" cette ligne de façon à ce que la date se trouve à la fin de cette première ligne.

11. Le compilateur refuse de compiler une routine avec un niveau élevé d'optimisation. ***Cause possible*** : Caprice du compilateur. ***Remède*** : utiliser un niveau de compilation moins élevé.
12. Très mauvaise convergence du calcul d'atmosphère. ***Cause possible*** : passage d'un point anguleux de la table d'opacité, ou modification de l'indice de grille du point d'indice $n_\star$. ***Remède*** : souvent, après quelques pas temporels, on s'écarte des points anguleux, et ces difficultés disparaissent. Utiliser des fonctions d'interpolation dérivables pour les tables d'opacité.
13. Très mauvaise convergence du calcul d'atmosphère. ***Cause possible*** : condition limite sur la pression qui ne peut être satisfaite au niveau de pécision élevé requis par l'intégration de l'atmosphère. ***Remède*** : utiliser la condition limite sur la densité.
14. L'évolution s'arrête *"normalement"* avant que l'âge requis ne soit atteint. ***Cause possible*** : signe erroné (*respt.* valeur trop forte, faible) du paramètre `log_teff` (*respt.* `x_stop, t_stop`) de la NAMELIST `nl_temps` du fichier `mon_modele.don`. ***Remède*** : modifier ces paramètres suivant les indications du § 3.15 (Page 33) précédent.
15. Mauvaise convergence de l'algorithme d'intégration temporelle pour ^{2}H, se produit vers la fin de la pré-séquence principale, se produit parfois conjointement aux sorties de table d'opacité. ***Cause possible*** : oscillations au moment de la mise à l'équilibre du deutérium. ***Remède*** : CESAM2k "force la convergence" qui, souvent au niveau de quelques 10^{-5}, n'est pas mauvaise.
16. Oscillations violentes de la quantité $A = \frac{1}{\Gamma_1}\frac{\partial \ln P}{\partial \ln R} - \frac{\partial \ln \rho}{\partial \ln R}$ d'un fichier `mon_modele-*.osc` *cf.* § D.1 (Page 239). Ces oscillations ont plusieurs causes décrites, *cf.* § 7.10.7 (Page 130). Bien que beaucoup d'efforts aient été consacrés à cette difficulté, jusqu'à présent, le moins mauvais remède consiste à supprimer "à la main" les couches incriminées.

    La diffusion des éléments chimique évite en grande partie ce phénomène. Lorsque celle-ci n'est pas requise et qu'une bonne représentation de la fréquence de Brunt-Väissälä est nécessaire, dans le fichier de données, *cf.* § 3.3 (Page 22), coder `'sa'` pour le type de précision, et introduire un peu de diffusion turbulente :

    ```
    /
    &NL_DIFF
    DIFFUSION=.TRUE.,
    NOM_DIFFM='diffm_0',
    NOM_DIFFT='difft_nu',
    D_TURB=10.d0,
    RE_NU=1.d0,
    NOM_FRAD='no_frad'
    /
    ```

    qui aura pour effet d'éliminer les disconinuités de la dérivée première de la composition chimique. Le coût à payer est un effort de calcul nettement plus important qu'avec `DIFFUSION=.FALSE.`. La figure 6.1 (Page 53) représente le profil de la fréquence de Brunt-Väissälä obtenu pour un modèle de $10M_\odot$

de séquence principale ($X_c = 0.35$), calculé avec la prodédure qui vient d'être décrite. Au cours de l'évolution le cœur convectif a régressé de $0.22 R_\star$ à $0.14 R_\star$.

17. Mauvaise convergence pour la restitution de l'atmosphère. Ce problème se produit parfois lors de la reprise d'un modèle. La difficulté est due au fait que l'atmosphère qui a été reprise en binaire n'est pas suffisamment proche de celle que CESAM2k désire calculer. ***Remède*** : supprimer le modèle d'atmosphère en binaire `mon_modele.atm`. Pour initialiser, CESAM2k utiliser l'atmosphère standard.
18. Mauvaise convergence, pas temporel tendant vers 0, et/ou oscillations de la vitesse angulaire au voisinage des limites zones radiatives / zones convectives. Les causes ne sont pas identifiées. Une conséquence est la génération d'instabilités de composition chimique, qui se transmettant à la densité, génèrent des coefficients très bruités pour les équations de la diffusion du moment cinétique. Une augmention du coefficient de diffusion turbulente `d_turb`, *cf.* § 3.15 (Page 33), permet, parfois, de faire disparaître d'anomalie.
19. Profil très bruité du taux de variation de poids moléculaire moyen Λ, *cf.* § 7.9.1 (Page 111), au voisinage des limites zones radiatives / zones convectives. Remède inconnu. Cette anomalie n'a, en général, pas de conséquence directe sur le profil de la vitesse angulaire.

Deuxième partie

DESCRIPTION DE CESAM2k

Chapitre 7

Numérisation de la physique

Dans ce chapitre, on expose la physique implantée dans CESAM2k conjointement aux méthodes numériques utilisées. Le but poursuivi est de donner le maximum d'éléments, afin de faciliter la compréhension tant de la physique implantée que des méthodes numériques utilisées. Certaines parties sont d'un niveau élémentaire, mais tout développement ultérieur de CESAM2k est à ce prix.

Les équations de l'évolution stellaire constituent un problème intégro-différentiel non linéaire aux dérivées partielles : c'est un problème aux limites de conditions initiales. Une seule dimension spatiale est prise en compte dans CESAM2k.

La méthode de résolution retenue, la méthode des "lignes" est classique, voir *e.g.* Henrici [1962]. Elle consiste, pour chaque pas temporel, à résoudre itérativement et de façon découplée, le problème de conditions initiales (problème de Cauchy) puis le problème aux limites (problème de Dirichlet) – une résolution couplée n'est envisageable que si le nombre d'espèces chimiques dont on doit suivre l'évolution temporelle est réduit, ce qui n'est pas le cas en général.

Le problème aux limites, non linéaire, est résolu par itérations Newton-Raphson qualifiée de "méthode d'Henyey" *et al.* [1959] en structure interne.

Le problème de valeurs initiales est un problème raide, car les diverses échelles de temps en présence diffèrent entre elles par plusieurs décades.

Une difficulté numérique est due à l'hypothèse physique suivant laquelle les mouvements convectifs homogénéisent les espèces chimiques et rigidifient la rotation. De ce fait la continuité de certaines fonctions inconnues est détruite aux limites zones radiatives / zones convectives.

7.1 Intégration par splines-collocation

On décrit dans ce paragraphe la méthode d'intégration numérique utilisée dans CESAM2k pour résoudre les problèmes aux limites que constituent les équations de l'équilibre quasi-statique et la restitution de l'atmosphère. Elle utilise des techniques et des algorithmes qui sont, pour la plupart, peu usités en évolution stellaire. Afin d'être accessible sans trop d'investissement en analyse numérique, mais avec le risque de plagier les manuels spécialisés, on donne un aperçu de la réflexion qui a conduit aux algorithmes retenus avant de décrire ces derniers.

Cette méthode consiste à développer les fonctions inconnues sur une base de polynômes par morceaux, les B-splines, et à écrire que ces développements satisfont les conditions limites et les équations différentielles en un nombre *ad–hoc* de points. Les fonctions de base ne satisfaisant pas les conditions limites, cette méthode est qualifiée de "quasi-spectrale" dans les ouvrages spécialisés.

7.1.1 Les B-splines normalisées

Etant donnée une partition de $[a, b] \subset : a = x_0 < x_1 < \ldots < x_n = b$, l'ensemble des polynômes par morceaux d'ordre m (degré +1), qui se raccordent en $x_1, x_2, \ldots, x_{n-1}$ est un espace vectoriel de dimension finie M, dont une base est celle des B-splines. Elle sera notée par la suite : $\{N_i^m\}_{i=1}^M$. Pour $m = 2$ et $n = 4$ ce sont les "fonctions chapeaux" de la Fig. 7.1 (Page 60) et les courbes en cloche pour $m = 4$ et $n = 7$. Les diverses définitions mathématiques de ces fonctions sont

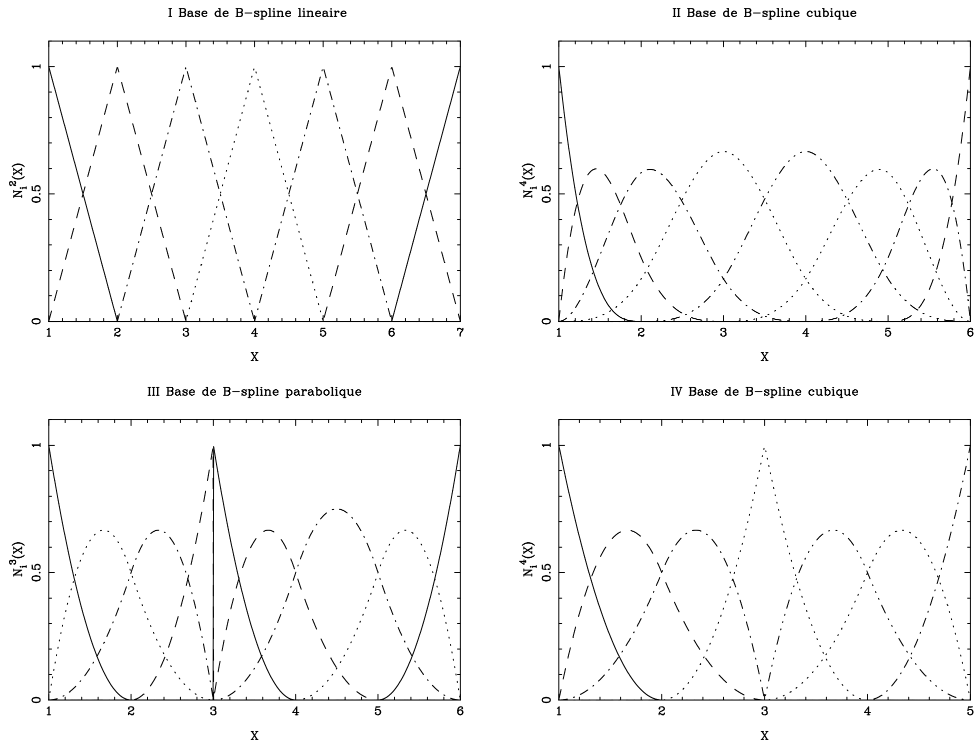

FIG. 7.1 – B-splines normalisées d'ordre m=2 pour n=4 points de grille équidistants.

abscondes : i-ème différence divisée de la fonction puissance translatée, par exemple. On les trouve dans C. de Boor [1978], p.108, Schumacker [1981], p.118, Trénoguine [1980], p.341, et dans de nombreux autres ouvrages. Ces fonctions de base, malgré leur algèbre compliquée se prètent très bien au calcul numérique car les calculs sont faciles, stables et efficaces. En particulier :

- les B-splines sont des fonctions à support borné,
- en un point de leur support il existe au plus m B-splines d'ordre m non nulles, une conséquence est la structure bande des systèmes linéaires apparaissant dans les calculs,
- une B-spline est toujours positive ou nulle,

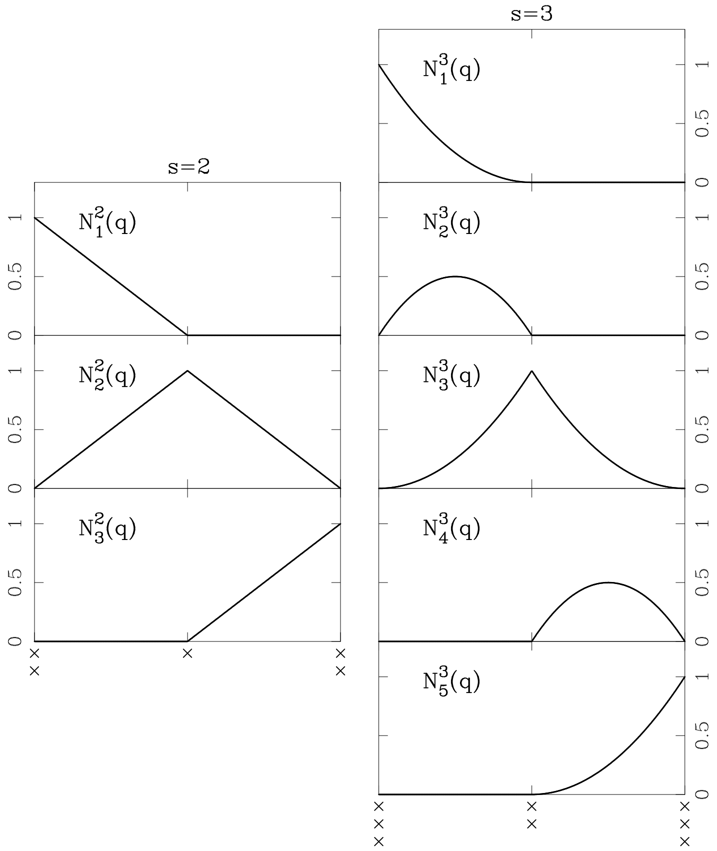

FIG. 7.2 – B-splines normalisées d'ordre s=2 (gauche) et s=3 (droite) calculées avec la base de C. de Boor pour n=3 points de grille équidistants. Les vecteurs nodaux sont schématisés par des croix disposées sous chaque graphe.

- la dérivée d'une B-spline est une combinaison linéaire de B-splines d'ordre immédiatement inférieur,
- il existe des algorithmes numériques efficaces et stables pour les calculs avec B-splines.

7.1.2 Résolution d'un problème différentiel

Le principe de l'utilisation des B-splines pour résoudre un problème différentiel est celui de toute méthode spectrale : les fonctions inconnues sont développées sur une base finie, en l'occurence celle des B-splines. Pour calculer les coefficients, on écrit que ces développements satisfont les équations différentielles et les conditions limites. Pour ce faire il y a une alternative :

- On satisfait les équations en un nombre *ad–hoc* de points, c'est la *collocation* qui est utilisée pour résoudre l'équilibre quasi-statique *cf.* § 7.2 (Page 66) et restituer l'atmosphère *cf.* § 7.3.2 (Page 76) .
- On impose à la base d'être orthogonale à son image par l'opérateur différentiel (*i.e.* au résidu), c'est la méthode des éléments finis (Galerkin) qui est utilisée pour résoudre les équations de diffusion des espèces chimiques, *cf.* § 7.8 (Page 96), et du moment cinétique *cf.* § 7.9 (Page 111).

Ainsi, pour le problème différentiel aux limites :

$$f''(x) = \Phi(x; f, f'), \quad \phi_1(x_1; f, f') = \alpha, \quad \phi_2(x_n; f, f') = \beta, \quad \text{sur } [x_1, x_n] \subset$$

où ϕ_1, ϕ_2 sont des fonctions et α, β des réels donnés. En utilisant la base $\{N_i^m\}_{i=1}^M$ de B-splines d'ordre $m \geq 4$, sur une partition de $[x_1, x_n]$ à préciser, on aura :

$$f(x) \simeq \sum_{i=1}^M f_i N_i^m(x), \quad f'(x) \simeq \sum_{i=1}^M f_i \frac{dN_i^m(x)}{dx}, \quad f''(x) \simeq \sum_{i=1}^M f_i \frac{d^2 N_i^m(x)}{dx^2}$$

Avec la méthode de collocation, les M coefficients f_i sont obtenus en écrivant que le développement de f vérifie les 2 conditions limites et les équations :

$$\sum_{i=1}^M f_i \frac{d^2 N_i^m(z_c)}{dx^2} = \Phi(z_c; f, f')$$

en $M - 2$ points $z_c \in]x_1, x_n[$.

La solution obtenue est en $O(||\Delta x||^m)$. Si on adopte un choix judicieux pour les points de collocation la solution est en $O(||\Delta x||^{2(m-1)})$: c'est la superconvergence de C. de Boor [1978], chap. 15.

Avec la méthode de Galerkin les coefficients f_i seraient, par exemple, obtenus en écrivant que le développement de f vérifie les M équations :

$$\sum_{i=1}^M f_i < \frac{d^2 N_i^m}{dx^2}.N_l^m >=< \Phi(x_l; f, f').N_l^m >, \quad 1 < l < M$$

où $< g.h >$ représente un produit scalaire à définir. Les conditions limites sont introduites par une intégration par parties, *cf.* Trénoguine [1980] ; Marchouk & Agochkov [1985].

Avec les deux méthodes, si Φ est linéaire, on obtient les coefficients f_i par résolution d'un système linéaire. Dans la cas de l'évolution stellaire, il est nécessaire d'utiliser une méthode itérative *e.g.* point fixe ou méthode de Newton-Raphson les équations étant non linéaires.

7.1.3 L'architecture de CESAM2*k*

L'algèbre des B-splines étant assez complexe, l'architecture de CESAM2*k* repose sur une séparation bien distincte entre l'espace physique et celui des fonctions de base. Les équations à résoudre sont écrites sous une forme proche de leur formulation dans l'espace physique. Dans l'espace des B-splines ces équations sont résolues pour les coefficients des fonctions de base. Des routines servent d'interfaces entre les deux espaces. Par exemple, les coefficients des équations de l'équilibre quasi-statique sont calculés dans l'espace physique par la routine `static_m`. Les équations sont résolues dans l'espace des splines par la routine `coll_qs`, la routine `resout` servant d'interface entre les deux espaces, voir la figure 1.1 (Page 7).

Les programmes d'exploitation des résultats opèrent le plus souvent dans l'espace physique.

7.1.4 Collocation pour un système non linéaire

On explicite la méthode spline/collocation pour le système de n_e équations différentielles ordinaires, non linéaires, du premier ordre, avec des conditions limites aux extrémités de l'intervalle de définition *e.g.* les équations de la structure interne. Ces systèmes s'écrivent formellement :

$$\mathcal{F}_j(x; \mathbf{f}, \mathbf{f}') = 0, \qquad x \in [x_1, x_n], \qquad \mathbf{f} = {}^T(f_1, f_2, \ldots, f_{n_e}) \qquad 1 \le j \le n_e$$

avec les conditions limites :

$$\alpha_q\left(\mathbf{f}(x_1)\right) = 0,\ \forall q \in Q,\ Q \subseteq J = \{1, 2, \ldots, n_e\}, \quad \beta_r\left(\mathbf{f}(x_n)\right) = 0,\ \forall r \in \mathcal{C}_J Q$$

la notation $\mathcal{C}_J Q$ signifiant le complémentaire de Q dans l'ensemble d'indices J. En remplaçant chaque composante f_j de $\mathbf{f}$, $1 \le j \le n_e$, par son développement sur la base des B-splines :

$$f_j(x) = \sum_{i=1}^{M} f_{i,j} N_i^m(x), \qquad f'_j(x) = \sum_{i=1}^{M} f_{i,j} \frac{dN_i^m(x)}{dx} \tag{7.1}$$

les inconnues sont désormais les réels $f_{i,j}$. Ils vérifient :

$$\begin{cases} \mathcal{F}_j\Big(x; \sum_{i=1}^M f_{i,1} N_i^m(x), \ldots, \sum_{i=1}^M f_{i,n_e} N_i^m(x), \sum_{i=1}^M f_{i,1} \frac{dN_i^m(x)}{dx}, \ldots, \\ \sum_{i=1}^M f_{i,n_e} \frac{dN_i^m(x)}{dx}\Big) = 0, \\ \forall x \in]x_1, x_n[,\ \forall j \in J \\ \alpha_q\Big(\sum_{i=1}^M f_{i,1} N_i^m(x_1), \ldots \sum_{i=1}^M f_{i,n_e} N_i^m(x_1)\Big) = 0,\ \forall q \in Q \\ \beta_r\Big(\sum_{i=1}^M f_{i,1} N_i^m(x_n), \ldots \sum_{i=1}^M f_{i,n_e} N_i^m(x_n)\Big) = 0,\ \forall r \in \mathcal{C}_J Q \end{cases}$$

Pour un système différentiel linéaire, les $f_{i,j}$ se rassemblent. Ils sont évalués en écrivant (7.1) en $M-1$ points de collocation $z_c \in]x_1, x_n[$ et en résolvant le système linéaire obtenu. Les équations de la structure interne ne sont pas linéaires. On utilise la méthode itérative de Newton-Raphson pour résoudre le système (7.5). Ce qui nécessite une solution initiale $\mathbf{f}^0$. En notant $\mathbf{f}^p$ la solution obtenue à l'itération $p \ge 0$, les équations linéarisées sont :

$$\begin{cases} \mathcal{F}_j\Big(z_c; \sum_{i=1}^M f_{i,1}^p N_i^m(z_c), \ldots, \sum_{i=1}^M f_{i,n_e}^p N_i^m(z_c), \sum_{i=1}^M f_{i,1}^p \frac{dN_i^m(z_c)}{dx}, \ldots, \\ \sum_{i=1}^M f_{i,n_e}^p \frac{dN_i^m(z_c)}{dx}\Big) = \\ = \sum_{k=1}^{n_e} \sum_{i=1}^M \Big(\frac{\partial \mathcal{F}_j}{\partial f_k} N_i^m(z_c) + \frac{\partial \mathcal{F}_j}{\partial f'_k} \frac{dN_i^m(z_c)}{dx}\Big) df_{i,k}^p,\ \forall j \in J,\ 1 \le c < M \\ \alpha_q\Big(\sum_{i=1}^M f_{i,1}^p N_i^m(x_1), \ldots \sum_{i=1}^M f_{i,n_e}^p N_i^m(x_1)\Big) = \\ \sum_{k=1}^{n_e} \frac{\partial \alpha_q}{\partial f_k} \sum_{i=1}^M N_i^m(x_1) df_{i,k}^p,\ \forall q \in Q, \\ \beta_r\Big(\sum_{i=1}^M f_{i,1}^p N_i^m(x_n), \ldots \sum_{i=1}^M f_{i,n_e}^p N_i^m(x_n)\Big) = \\ \sum_{k=1}^{n_e} \frac{\partial \beta_r}{\partial f_k} \sum_{i=1}^M N_i^m(x_n) df_{i,k}^p,\ \forall r \in \mathcal{C}_J Q \end{cases} \tag{7.2}$$

Les sommes sur i ne portent, en fait, que sur m indices puisqu'en un point x il y a, au plus, m B-splines non nulles. Le système (7.2) est linéaire pour les corrections $df_{i,k}^p$. On aura :

$$f_{i,k}^{p+1} = f_{i,k}^p - df_{i,k}^p, \qquad 1 \le i \le M, \quad 1 \le k \le n_e$$

Sans diffusion, à cause du mélange convectif, aux limites zones radiatives / zones convectives la composition chimique présente une discontinuité, en résulte une discontinuité de la densité. Avec diffusion, les discontinutés n'affectent que les dérivées premières. Ces discontinuités ont conduit au choix de la pression comme variable dépendante au lieu de la densité. De plus, l'utilisation de cette dernière nécessiterait l'évaluation des dérivées partielles $(\partial P/\partial \rho)_{T,X}$, $(\partial P/\partial T)_{\rho,X}$ et $(\partial P/\partial X)_{T,\rho}$ dans les deux premières équations du système (7.3), et non seulement les dérivées secondes "thermodynamiques" :

$$\left(\frac{\partial^2 P}{\partial T^2}\right)_{\rho,X}, \quad \left(\frac{\partial^2 P}{\partial \rho^2}\right)_{T,X}, \quad \left(\frac{\partial^2 P}{\partial \rho \partial T}\right)_{X}, \quad \left(\frac{\partial^2 P}{\partial \rho \partial X}\right)_{T}, \quad \left(\frac{\partial^2 P}{\partial T \partial X}\right)_{\rho}$$

Les équations de l'équilibre quasi-statique sont intégrées avec un schéma supposant la continuité de chaque variable dépendante avec, éventuellement, une discontinuité de la dérivée première, *cf.* § 7.1.5 (Page 64). Sans diffusion, en raison de sa nature discontinue, au voisinage des limites zones radiatives / zones convectives, le profil de γ présentera des oscillations, toutefois sans importance car, sans diffusion, γ n'est pas couplé aux autres variables. Il n'en est pas de même avec diffusion, mais alors γ est continue, non dérivable aux limites zones radiatives / zones convectives, situation prise en compte par le schéma d'intégration.

7.1.5 Choix de la base de la collocation

Ainsi qu'ils sont définis dans C. de Boor [1978], p. 280, un choix particulier de la base et des points de collocation permet d'obtenir un ordre de précision supérieur à celui des fonctions d'interpolation c'est la *superconvergence*. On se donne l'ordre des B-splines m et la grille des points de raccord. Les morceaux de polynômes se raccordent en ces points en assurant la continuité de la B-spline et de ses dérivées jusqu'à un ordre inférieur d'une unité à celui du système différentiel. Dans CESAM2k on utilise la grille équidistante $\{1, 2, \dots n\}$, l'ordre des équations différentielles est $r = 1$ et l'ordre des B-splines est fixé par le type de précision indiqué dans le fichier de données *cf.* § 3.3 (Page 22), ou, éventuellement dans un fichier `reglages` *cf.* § 4.8 (Page 43).

Entre deux points de la grille, les variables dépendantes sont interpolées par des morceaux de polynômes d'ordre $m + r = 2$ *i.e.* droites, ou 3 *i.e.* paraboles. Les morceaux de polynômes se raccordent sur les points de grille par continuité, leurs dérivées premières pouvant être, au besoin, différentes. Cette particularité est exploitée pour décrire correctement le profil de la densité aux limites zones radiatives / zones convectives, *cf.* § 7.2.1 (Page 67).

La figure 3 représente cette base pour $n = 3$. Aux n points de la grille, la solution est d'ordre $2m = 2$ ou 4 ; cette superconvergence est due à cette disposition particulière des points de collocation. Entre les points de raccord la solution est d'ordre $m+r = 2$ ou 3 ; la superconvergence ne peut donc être exploitée directement en tout point. S'il est nécessaire de connaître la solution en des points autres que ceux définis par la grille, pour exploiter la superconvergence, il faut faire une interpolation d'ordre $2m$[1].

[1]L'expérience des calculs a montré que l'exploitation de la superconvergence était délicate à cause des discontinuités dues aux déplacements des limites zone radiative / zone convective.

Dans ces conditions, la dimension de la base de B-splines est $M = (n-1)m + r$. C'est le nombre des coefficients $f_{i,j}$ à calculer pour chaque inconnue (indice j). Il faut rassembler un nombre égal d'équations de conditions et il y a autant de conditions limites que d'inconnues. Il faut donc $(n-1)m$ points de collocation c'est à dire m points de collocation entre deux points de raccord. Ces points sont disposés aux abscisses des zéros du polynôme de Legendre de degré m, défini sur chaque intervalle ramené à $[-1, 1]$; soit ± 0.57735 et 0 pour $m = 2$. La superconvergence résulte de cette disposition. Ces points fixés, on épargne du temps calcul en y déterminant, une fois pour toutes, les valeurs des B-splines et de leurs dérivées premières.

CESAM2k utilise l'algorithme de C. de Boor [1978], selon la version de Schumaker [1981], chap. 5, pour calculer la valeur de toutes les B-splines non identiquement nulles en un point quelconque. Cet algorithme est basé sur un calcul de différences divisées pour une interpolation de Hermite, c'est l'objet des routines `bval0, bval1, bvald`, *cf.* § 9.2 (Page 215). Sa disposition pratique repose sur la construction d'une table d'abscisses où certaines sont répétées plusieurs fois. Elle constitue le ***vecteur nodal*** construit dans la routine `coll`, *cf.* § 9.2.7 (Page 216).

En chaque point de collocation, l'ordre d'interpolation étant $m+r = 3$ ou $m+r = 2$, il y a 3 ou 2 B-splines non nulles. Chacune des Eq. (7.5) fait intervenir 3 ou 2 coefficients non identiquement nuls par inconnue et, selon qu'on utilise $m = 1$ ou $m = 2$ on dispose 1 ou 2 points de collocation entre deux points de raccord, la structure du jacobien de la méthode de Newton-Raphson est la suivante puisque les fonctions de base ont un support borné :

$$m = 1 \ : \ \begin{pmatrix} \times & \times & & & & & \\ \circ & \circ & & & & & \\ & \times & \times & & & & \\ & & \cdot & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & \times & \times & \\ & & & & & \times & \times \\ & & & & & \circ & \circ \end{pmatrix}$$

$$m = 2 \ : \ \begin{pmatrix} \times & \times & \times & & & & & & & \\ \times & \times & \times & & & & & & & \\ \circ & \circ & \circ & & & & & & & \\ & & \times & \times & \times & & & & & \\ & & \times & \times & \times & & & & & \\ & & & & \cdot & & & & & \\ & & & & & \cdot & & & & \\ & & & & & & \cdot & & & \\ & & & & & & \times & \times & \times & & \\ & & & & & & \times & \times & \times & & \\ & & & & & & & & \times & \times & \times \\ & & & & & & & & \times & \times & \times \\ & & & & & & & & \circ & \circ & \circ \end{pmatrix}$$

à chaque point de collocation correspond une ligne de $\times$. Chacun d'eux[2] représente une matrice carrée dont l'ordre est le nombre d'inconnues $n_e = 7$, $n_e = 8$ avec pression turbulente. Par exemple, avec $m = 2$, à chaque intervalle entre deux points de raccord, correspond un bloc de $m = 2$ lignes de $(m+r) \times = (2+1) \times = 3 \times$. Deux blocs consécutifs se raccordant par $r = 1$ colonne de blocs. Les lignes correspondant aux conditions limites sont représentées par des $\circ$. Ce sont des matrices de format $3 \times n_e$, puisqu'il y a trois conditions limites à chaque extrémité de l'intervalle $[1, n]$ d'intégration.

Pour la structure interne, le système linéaire est formé et résolu dans le sous-programme `coll_qs` *cf.* § 8.18 (Page 151) à partir des coefficients calculés dans `static_m`, *cf.* § 8.123 (Page 204), ou `static_r`, *cf.* § 8.123 (Page 204). Pour la restitution de l'atmosphère, il s'agit respectivement des routines `coll_atm`, *cf.* § 8.17 (Page 150) et `eqatm`, *cf.* § 8.45 (Page 162).

On peut remarquer qu'il suffit de garder en mémoire des lignes de longueur $(m+r)n_e = (2+1)6 = 18$ pour résoudre le sytème. Il comporte $n_e((n-1)m+r) = 6((150-1)2+1) = 1794$ lignes pour $n = 150$. Cette remarque est exploitée dans la routine `gausdp_g` servant à résoudre le système par élimination de Gauss avec pivot partiel, *cf.* § 9.1 (Page 213).

S'il n'est pas tenu compte de la pression turbulente une équation est omise.

7.2 Equations de la structure interne

CESAM2k résout les équations de la structure interne sous une forme assez éloignée de leur expression classique. Dans ce paragraphe, on établit les relations utilisées, puis on donne le changement de variable permettant d'adapter automatiquement le réseau à l'évolution des variations des inconnues, enfin on décrit la méthode utilisée pour fixer un point de la grille de discrétisation à chaque limite entre une zone radiative et une zone convective.

Avec, comme variable indépendante lagrangienne, la masse M contenue à l'intérieur d'une sphère de rayon R, les équations de l'évolution stellaire sont *e.g.* (Kippenhahn *et al.* [1991]) :

$$\left\{\begin{array}{l} \frac{\partial P}{\partial M} = -\frac{GM}{4\pi R^4} + \frac{\Omega^2}{6\pi R} \\ \frac{\partial T}{\partial M} = \frac{\partial P}{\partial M}\frac{T}{P}\nabla \\ \frac{\partial R}{\partial M} = \frac{1}{4\pi R^2 \rho} \\ \frac{\partial L}{\partial M} = \epsilon - \epsilon_{\rm G} = \epsilon - \frac{\partial U}{\partial t} + \frac{P}{\rho^2}\frac{\partial \rho}{\partial t} \\ P = P_{\rm gaz} + P_{\rm tur} \\ \frac{\partial X_{\rm i}}{\partial t} = -\frac{\partial F_i}{\partial M} + \Psi_i(P_{\rm gaz}, T; \mathcal{X}) \\ 1 \le i \le n_{\rm elem}. \end{array}\right. \quad \left\{\begin{array}{l} \text{conditions limites :} \\ R(0,t) = 0,\ L(0,t) = 0, \\ P(M_{\rm b}, t) = P_{\rm b}(L, R), \\ P_{\rm gaz}(M_{\rm b}, t) = P_{\rm gaz\ b}(L, R), \\ T(M_{\rm b}, t) = T_{\rm b}(L, R), \\ \text{conditions initiales :} \\ X_i(M, 0) = X_{i,0}, \\ i = 1, \ldots, n_{\rm elem}, \\ P(M, 0) = P_0(M), \\ P_{\rm gaz}(M, 0) = P_{\rm gaz\ 0}(M), \\ T(M, 0) = T_0(M), \\ R(M, 0) = R_0(M), \\ L(M, 0) = L_0(M), \\ 0 \le M \le M_{\rm b}. \end{array}\right. \tag{7.3}$$

[2] Si on ne tient pas compte de la pression turbulente.

les notations classiques ont été utilisées ; R : rayon (distance au centre de l'étoile), $P \equiv P_{\rm tot}$: pression totale, $P_{\rm gaz}$: pression gazeuse, $P_{\rm tur}$: pression turbulente, ρ : densité, Ω : vitesse angulaire, G : constante de la gravitation, T : température, $\nabla \equiv \left(\frac{\partial \ln T}{\partial \ln P}\right)_t$: gradient, L : luminosité, $\epsilon = \epsilon_{\rm nu} - \frac{\partial \epsilon_\Omega}{\partial t} - \epsilon_\nu$: débit d'énergie résultant des réactions nucléaires, de la dissipation locale du moment cinétique *cf.* § 7.6 (Page 92), et des pertes par neutrinos, $\epsilon_{\rm G}$ est l'énergie gravifique, Eq. 7.4.2 (Page 83), M : masse incluse dans la sphère de rayon R, t : temps, X_i : abondance de l'élément chimique i, F_i : flux diffusif pour l'élément chimique i, $\mathcal{X} = \{X_i\}$: vecteur de composition chimique, Ψ_i : taux de variation de l'abondance de l'élément chimique i par les réactions thermonucléaires, $n_{\rm elem}$: nombre d'espèces chimiques considérées. $M_{\rm b}(R, L)$, $P_{\rm b}(L, R)$, $P_{\rm gaz_b}(L, R)$ et $T_{\rm b}(L, R)$ sont quatre fonctions décrivant respectivement les variations de la masse, de la pression totale, de la pression gazeuse et de la température, en fonction du rayon et de la luminosité à la limite extérieure de l'enveloppe qui coïncide avec la base de l'atmosphère. Les 5 fonctions : $P_0(M)$, $P_{\rm gaz\ 0}(M)$, $T_0(M)$, $R_0(M)$ et $L_0(M)$ décrivent, en fonction de la masse, respectivement la pression totale, la pression gazeuse, la température, le rayon et la luminosité du modèle d'âge zéro.

Cette formulation suppose la symétrie sphérique et ne tient pas compte du champ magnétique, et la rotation n'y intervient que par la gravité effective. Les équations décrivant la diffusion des éléments chimiques et du moment cinétique seront respectivement décrites au § 7.8 (Page 96) et § 7.9 (Page 111).

7.2.1 Discontinuités de la densité

L'échelle de temps de retournement des éléments convectifs étant petite devant l'échelle de temps nucléaire il est supposé que la convection homogénéise les zones convectives, *cf.* § 7.9.15 (Page 120). La composition chimique, et donc la densité, sont ***discontinues*** aux limites zones radiatives / zones convectives[3]. CESAM2*k* utilise des vecteurs nodaux pour les diverses intégrations (structure, composition chimiques, moment cinétique) construits de façon telle que les points où sont formées les équations ne coïncident pas avec les points de discontinuité, ainsi ces dernières sont implicitement prises en compte.

7.2.2 Variables d'intégration

Les variables naturelles par rapport auxquelles sont écrites les équations (7.3) ne sont pas bien adaptées au calcul numérique. En particulier, les dérivées de la pression et du rayon ont des singularités en $M \equiv 0$, *i.e.* au centre.

Ces singularités disparaissent en utilisant le rayon R comme variable indépendante. La condition limite externe devient alors une condition de limite libre, puisque le rayon total n'est pas fixe. Bien que cette limite libre ne soit pas une difficulté pour la méthode numérique utilisée dans CESAM2*k*, le rayon n'est pas, pour autant, le meilleur choix pour la variable indépendante ; on peut en effet montrer (Morel [1997]) que les singularités sont levées et qu'une utilisation optimale de l'approximation numérique est obtenue lorsqu'on utilise respectivement pour la masse, le rayon et la

[3]Quand la diffusion des éléments chimiques est prise en compte, la composition chimique et la densité sont ***continues non dérivables*** en ces limites.

luminosité les variables suivantes : $M^{2/3}$, R^2 et $L^{2/3}$. Il est apparu pratique d'utiliser des variables normalisées aux valeurs solaires :

$$\xi = \ln P,\ \xi_g = \ln P_{\text{gaz}},\ \eta = \ln T,\ \mu = \left(\frac{M}{M_\odot}\right)^{\frac{2}{3}},\ \zeta = \left(\frac{R}{R_\odot}\right)^2,\ \lambda = \left(\frac{L}{L_\odot}\right)^{\frac{2}{3}}, \gamma = \ln\rho$$

Avec ces nouvelles variables, le système (7.3) devient :

$$\begin{cases}
\frac{\partial \xi}{\partial \mu} = \left[-\frac{3G}{8\pi}\left(\frac{M_\odot}{R_\odot^2}\right)^2 + \frac{M_\odot}{4\pi R_\odot}\sqrt{\frac{\mu}{\zeta}}\Omega^2\right]\exp(-\xi)\left(\frac{\mu}{\zeta}\right)^2 \\
\frac{\partial \eta}{\partial \mu} = \frac{\partial \xi_t}{\partial \mu}\nabla \\
\frac{\partial \zeta}{\partial \mu} = \frac{3}{4\pi}\frac{M_\odot}{R_\odot^3}\frac{1}{\rho}\left(\frac{\mu}{\zeta}\right)^{\frac{1}{2}} \\
\exp(\xi) = \exp(\xi_g) + P_{\text{tur}} \\
\gamma = \ln\rho(\exp(\xi_g), \exp(\eta), X_i) \\
\frac{\partial \lambda}{\partial \mu} = \frac{M_\odot}{L_\odot}\left(\frac{\mu}{\lambda}\right)^{\frac{1}{2}}\left[\epsilon - \frac{\partial U}{\partial t} + \frac{P}{\rho^2}\frac{\partial \rho}{\partial t}\right] \\
\frac{\partial X_i}{\partial t} = -\frac{2}{3M_\odot\sqrt{\mu}}\frac{\partial F_i}{\partial \mu} + \Psi_i(\xi_t, \eta; \mathcal{X}),\ 1 \le i \le n_{\text{elem}}
\end{cases}$$

$$\begin{cases}
\text{conditions limites} \\
\zeta(1,t) = 1,\ \lambda(1,t) = 1 \\
\xi(\mu_{\text{ext}}, t) = \ln\left(P_{\text{ext}}\left(\sqrt{\lambda}^3 L_\odot, \sqrt{\zeta}R_\odot\right)\right) \\
\xi_{\text{g}}(\mu_{\text{ext}}, t) = \xi(\mu_{\text{ext}}, t) \\
\eta(\mu_{\text{ext}}, t) = \ln\left(T_{\text{ext}}\left(\sqrt{\lambda}^3 L_\odot, \sqrt{\zeta}R_\odot\right)\right) \\
\gamma = \ln\rho(P_{\text{ext}}, T_{\text{ext}}, X_i) \\
\text{conditions initiales} \\
X_i(\mu, 0) = X_{i,0},\ i = 1, \dots, n_{\text{elem}} \\
\xi(\mu, 0) = \xi_0(\mu),\ \xi_{\text{g}}(\mu, 0) = \xi_{\text{g}\,0}(\mu), \\
\eta(\mu, 0) = \eta_0(\mu),\ \gamma(\mu, 0) = \gamma_0(\mu) \\
\zeta(\mu, 0) = \zeta_0(\mu),\ \lambda(\mu, 0) = \lambda_0(\mu), \\
1 \le \mu \le \mu_{\text{ext}}
\end{cases}$$

Au centre, la singularité des gradients est levée :

$$\lim_{\mu\to 0}\left|\frac{\partial \xi}{\partial \mu}\right| = \frac{G}{2}\left(\frac{4\pi M_\odot^2}{3}\right)^{\frac{1}{3}}\exp\left(-\xi(0,t)\right)\rho(0,t)^{\frac{4}{3}} < \infty,$$

$$\lim_{\mu\to 0}\left|\frac{\partial \zeta}{\partial \mu}\right| = \left(\frac{3M_\odot}{4\pi R_\odot^3}\right)^{\frac{2}{3}}\rho^{-\frac{2}{3}}(0,t) < \infty,$$

et, de plus :

$$\lim_{\mu\to 0}\left|\frac{\mu}{\lambda}\right| = \lim_{\mu\to 0}\left|\frac{\partial \lambda}{\partial \mu}\right| = \frac{M_\odot}{L_\odot}\left[\epsilon - T\left(\frac{\partial S}{\partial t}\right)_\mu\right],$$

$$\lim_{\mu\to 0}\frac{\mu}{\zeta} = \left(\frac{4\pi}{3}\right)^{\frac{2}{3}}\left(\frac{R_\odot^2}{M_\odot}\right)^{\frac{2}{3}}\rho^{\frac{2}{3}},$$

quantités dérivables une fois au moins. Sans l'exposant 2/3 qui affecte la luminosité, le second membre de l'équation d'énergie eut été :

$$\frac{\partial \lambda}{\partial \mu} = \frac{M_\odot}{L_\odot}\mu^{\frac{1}{2}}\left[\epsilon - \frac{\partial U}{\partial t} + \frac{P}{\rho^2}\frac{\partial \rho}{\partial t}\right]$$

la racine carrée, mal représentée par un polynôme, au voisinage de $\mu = 1$, entrainant une instabilité numérique. L'exposant $2/3$ permet donc une meilleure représentation numérique de la luminosité au voisinage du centre, mais interdit à cette variable dépendante de devenir négative, ce qui, dans certaines phases de l'évolution, peut devenir rédhibitoire. Lorsque cette éventualité peut se présenter, bien que moins satisfaisant numériquement, il est nécessaire d'utiliser le rayon comme variable indépendante eulérienne et, dans ce cas, l'ensemble de variables utilisé par CESAM2k est :

$$\xi = \ln P,\ \xi_{\rm g} = \ln P_{\rm gaz},\ \eta = \ln T,\ \mu = \left(\frac{M}{M_\odot}\right),\ \zeta = \left(\frac{R}{R_\odot}\right),\ \lambda = \left(\frac{L}{L_\odot}\right),\ \gamma = \ln \rho.$$

7.2.3 Adaptation de la discrétisation

Les zones dans lesquelles les solutions ont une variation rapide se déplacent au cours de l'évolution, si bien que le réseau initial devient plus ou moins rapidement inadapté. Pour ne pas dégrader la précision, il faut alors resserrer les nœuds dans les régions à variations fortes, les relacher sur les plateaux. Deux solutions peuvent être apportées à ce problème numérique difficile :

1. on ajoute/retire une ou plusieurs couches lorsque l'incrément d'une variable est trop/pas assez important (Kippenhahn *et al.* [1968]).
2. on dispose les nœuds automatiquement de façon à satisfaire certains critères (Eggleton [1971], Press *et al.* [1986]).

CESAM2k utilise la seconde solution. L'idée consiste à effectuer un changement de variable. On utilise comme variable dépendante une fonction, à définir, des variables dépendantes dont on désire contrôler les variations, fonction à laquelle on impose une variation constante d'un nœud de la grille au suivant. Dans un premier temps le nombre total de nœuds est fixé. Ceux–ci sont disposés automatiquement par le calcul aux abscisses *ad–hoc*. La masse, variable indépendante devient une variable dépendante dont la valeur n'est connue qu'à l'issue de l'intégration. A cette fin on définit la fonction d'espacement $Q(\mu, t)$ à partir de la connaissance que l'on a, *à priori*, de la solution. Par exemple, pour la structure interne, on pourrait[4] prendre : $Q(\mu, t) = \ln P$. On cherche à l'instant t une répartition des abscisses $\mu_i, i = 1, \dots, n$ telle que l'incrément de la fonction d'espacement soit constant d'un nœud à l'autre de la grille. On cherchera donc à réaliser :

$$Q(\mu_i, t) - Q(\mu_{i+1}, t) = Cte(t), \quad i = 1, \dots, n-1 \tag{7.4}$$

Celà impose une monotonie stricte à la fonction d'espacement $Q(\mu, t)$. Le nombre de couches n étant donné, pour t fixé, on introduit la fonction "indice" $q(\mu, t)$ appliquant l'intervalle de variation de μ : $[0, \mu_{\rm ext}]$ sur $[1, n] \subset \mathbf{R}$. La condition Eq. 7.4 (Page 69) s'écrit alors de façon équivalente :

$$\left(\frac{\partial Q}{\partial q}\right)_t = Cte(t) = \psi(t) \Longleftrightarrow \left(\frac{\partial^2 Q}{\partial q^2}\right)_t = \left(\frac{\partial \psi}{\partial q}\right)_t = 0$$

[4] Ce qui équivaut à prendre la pression comme variable indépendante.

et, pour les conditions limites : $\mu = 0$ à $q = n$ et $\mu = \mu_{\text{ext}}$ à $q = n$, puisque la couche d'indice 1 correspond au centre, et celle d'indice n à la surface.

Pour t fixé, le changement de variable : $\mu \to q(\mu, t)$ permet d'écrire :

$$\left(\frac{\partial Q}{\partial q}\right)_t = \left(\frac{\partial Q}{\partial \mu}\right)_t \left(\frac{\partial \mu}{\partial q}\right)_t = \theta \left(\frac{\partial \mu}{\partial q}\right)_t$$

où $\theta(\mu, t) = \left(\frac{\partial Q}{\partial \mu}\right)_t$ est calculé à partir de l'expression de la fonction d'espacement $Q(\mu, t)$. Il y a donc deux fonctions inconnues supplémentaires : $\psi(t) = \left(\frac{\partial Q}{\partial q}\right)_t$ et $\mu(q, t)$, solutions du système différentiel du premier ordre avec conditions limites :

$$\left(\frac{\partial \mu}{\partial q}\right)_t = \frac{\psi}{\theta}, \qquad \left(\frac{\partial \psi}{\partial q}\right)_t = 0, \text{ avec } : \begin{cases} q = 1, \ \mu = 1 \\ q = n, \ \mu = \mu_{\text{ext}} \end{cases}$$

Ces équations sont résolues simultanément avec le système des équations (7.3) écrites par rapport aux variables q et t ; pour ce faire, on dispose des relations suivantes, où $f(\mu,\ t)$ désigne une variable dépendante quelconque :

$$\left(\frac{\partial f}{\partial t}\right)_\mu = \left(\frac{\partial f}{\partial t}\right)_q - \left(\frac{\partial f}{\partial \mu}\right)_t \left(\frac{\partial \mu}{\partial t}\right)_q, \qquad \left(\frac{\partial f}{\partial \mu}\right)_t = \left(\frac{\partial f}{\partial q}\right)_t \Big/ \left(\frac{\partial \mu}{\partial q}\right)_t.$$

Les abscisses utilisées sont 1, 2, ... , n, on dispose donc d'une **grille équidistante**.

7.2.4 Modification du nombre total de couches

On a supposé jusqu'à présent que le nombre total de couches était fixé une fois pour toutes, ce qui constitue évidemment une contrainte, puisque pour une phase donnée de l'évolution, il y a le risque de travailler avec une discrétisation trop fine (*respt.* grossière) avec comme conséquence des calculs inutiles (*respt.* imprécis). Avec une augmentation (*respt.* diminution) du nombre de couches, le saut[5] $\psi(t)$ de la fonction de répartition d'une couche à la suivante diminue (*respt.* augmente), si bien que d'un pas temporel au suivant, il est possible d'ajuster le nombre de couches afin de maintenir $\psi(t)$ dans un intervalle fixé. Cette disposition correspond exactement au sens physique que l'on peut attribuer à la fonction de répartition qui est de maintenir dans un intervalle donné les variations de certaines grandeurs physiques.

Alors $\psi(t) = \left(\frac{\partial Q}{\partial q}\right)_\mu$ et $\theta(\mu, t) = \left(\frac{\partial Q}{\partial \mu}\right)_t$, $\Rightarrow \frac{\psi}{\theta} = (\frac{\partial \mu}{\partial q})_t$ et les équations (7.3)

[5] $\psi(t)$ est soit positif, soit négatif, ici il est supposé positif.

deviennent :

$$
\left\{
\begin{array}{l}
\frac{\partial \xi}{\partial q} = \left[-\frac{3G}{8\pi} \left(\frac{M_\odot}{R_\odot^2} \right)^2 \left(\frac{\mu}{\zeta} \right)^2 + \frac{M_\odot}{4\pi R_\odot} \sqrt{\frac{\mu}{\zeta}} \Omega^2 \right] \exp(-\xi) \frac{\psi}{\theta} \\
\frac{\partial \eta}{\partial q} = \frac{\partial \xi}{\partial q} \nabla \\
\frac{\partial \zeta}{\partial q} = \frac{3}{4\pi} \frac{M_\odot}{R_\odot^3} \frac{1}{\rho} \left(\frac{\mu}{\zeta} \right)^{\frac{1}{2}} \frac{\psi}{\theta} \\
\exp(\xi) = \exp(\xi_g) + P_{\rm tur} \\
\gamma = \ln \rho(\exp(\xi_g), \exp(\eta), X_i) \\
\frac{\partial \lambda}{\partial q} = \frac{M_\odot}{L_\odot} \left(\frac{\mu}{\lambda} \right)^{\frac{1}{2}} \left[\epsilon - \frac{\partial U}{\partial t} + \frac{P}{\rho^2} \frac{\partial \rho}{\partial t} \right] \frac{\psi}{\theta} \\
\frac{\partial \mu}{\partial q} = \frac{\psi}{\theta} \\
\frac{\partial \psi}{\partial q} = 0 \\
\frac{\partial X_i}{\partial t} = -\frac{2}{3M_\odot \sqrt{\mu}} \frac{\partial F_i}{\partial \mu} + \Psi_i(\xi_t, \eta; \mathcal{X}),\ 1 \le i \le n_{\rm elem} \\
i = 1, \ldots, n_{\rm elem}
\end{array}
\right. \tag{7.5}
$$

$$
\left\{
\begin{array}{l}
\text{conditions limites} \\
\mu(n,t) = \mu_{\rm ext} \left(\sqrt{\lambda}^3(n,t) L_\odot, \sqrt{\zeta(n,t)} R_\odot \right), \\
\xi(n,t) = \ln \left(P_{\rm ext} \left(\sqrt{\lambda}^3(n,t) L_\odot, \sqrt{\zeta(n,t)} R_\odot \right) \right), \\
\xi_{\rm g}(n,t) = \xi(n,t), \\
\eta(n,t) = \ln \left(T_{\rm ext} \left(\sqrt{\lambda}^3(n,t) L_\odot, \sqrt{\zeta(n,t)} R_\odot \right) \right), \\
\gamma(n,t) = \ln \rho(P_{\rm ext}, T_{\rm ext}, X_i) \\
\zeta(1,t) = 0,\ \lambda(1,t) = 0,\ \mu(1,t) = 0. \\
\text{conditions initiales} \\
X_i(q,0) = X_{i,0},\ i = 1, \ldots, n_{\rm elem} \\
\xi(q,0) = \xi_0(q),\ \xi_{\rm g}(q,0) = \xi_{\rm g\ 0}(q),\ \eta(q,0) = \eta_0(q), \\
\zeta(q,0) = \zeta_0(q),\ \lambda(q,0) = \lambda_0(q),\ \gamma(q,0) = \gamma_0(q) \\
1 \le q \le n.
\end{array}
\right.
$$

Ces équations de l'espace physique sont formées dans la routine `static_m`, *cf.* § 8.123 (Page 204). La fonction d'espacement Q doit être strictement monotone ; il faut donc qu'elle ne soit composée que de fonctions variant dans le même sens. Il va de soi que son expression doit être la plus simple possible, et il n'y a pas nécessaire obligation de la contraindre avec des "variables secondaires" telle que $P_{\rm gaz}$ ou Ω. Puisque ξ et η varient en sens inverse de ζ, λ et μ, quand il n'y a aucune raison de favoriser une variable plutôt qu'une autre, formellement, il conviendrait d'utiliser :

$$Q(\mu,t) = \frac{\xi}{\Delta\xi} + \frac{\eta}{\Delta\eta} - \frac{\zeta}{\Delta\zeta} - \frac{\lambda}{\Delta\lambda} - \frac{\mu}{\Delta\mu},\ \Delta\xi = \xi(n) - \xi(1),$$
$$\Delta\eta = \eta(n) - \eta(1),\ \Delta\zeta = \zeta(n) - \zeta(1),\ \Delta\lambda = \lambda(n) - \lambda(1),\ \Delta\mu = \mu(n) - \mu(1).$$

On aurait donc :

$$
\begin{array}{l}
\theta(\mu,t) = \left(\frac{\partial Q}{\partial \mu} \right)_t = \frac{1}{\Delta\xi} \frac{3G}{8\pi} \left(\frac{M_\odot}{R_\odot^2} \right)^2 \exp(-\xi) \left(\frac{\mu}{\zeta} \right)^2 \left(1 + \nabla \frac{\Delta\xi}{\Delta\eta} \right) - \\
- \frac{1}{\Delta\zeta} \frac{3}{4\pi} \frac{M_\odot}{R_\odot^3} \frac{1}{\rho} \left(\frac{\mu}{\zeta} \right)^{\frac{1}{2}} - \frac{1}{\Delta\lambda} \frac{M_\odot}{L_\odot} \left(\frac{\mu}{\lambda} \right)^{\frac{1}{2}} \epsilon - 1.
\end{array} \tag{7.6}
$$

et des relations similaires lorsque le rayon est utilisé comme variable indépendante. Les facteurs de répartition $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$, $\Delta\lambda$ et $\Delta\mu$ peuvent être, ou bien fixés une fois pour toutes ou bien, ajustés à l'issue de chaque pas temporel de façon à suivre l'évolution.

Si cette forme "complète" de la fonction de répartition a été utilisée pour les versions 0, 1 et 2 de CESAM2*k*, il s'est avéré par la suite que $Q \equiv a\xi + b\mu$ (*respt.* $Q \equiv a\xi + b\zeta$) avec les variables lagrangiennes (*respt.* eulériennes) semblait être la forme la plus simple, la plus robuste et de fait, la plus efficace. Les valeurs $a = -1$ et $b \sim 15$ des facteurs de répartition, définies dans la routine `cesam`, *cf.* § 8.10 (Page 147), selon le type de précision requis, donnent satisfaction.

L'approximation initiale de $\psi(t_0) = \left(\frac{\partial Q}{\partial q}\right)_t$ s'obtient en dérivant numériquement $Q(\mu, t)$ (obtenu à partir de la solution initiale provisoire) par rapport à la nouvelle variable indépendante : la fonction de discrétisation $q(\mu, t)$.

Avec la variable indépendante eulérienne on a : $\psi(t) = \left(\frac{\partial Q}{\partial q}\right)_\zeta$ et $\theta(\zeta, t) = \left(\frac{\partial Q}{\partial \zeta}\right)_t$ et le problème différentiel s'écrit :

$$\left\{\begin{array}{l}
\frac{\partial \xi}{\partial q} = \left[-\frac{GM_\odot}{R_\odot}\frac{\mu}{\zeta^2} + \frac{2R_\odot}{3}\zeta\Omega^2\right]\rho\exp(-\xi)\frac{\psi}{\theta} \\
\frac{\partial \eta}{\partial q} = \frac{\partial \xi}{\partial q}\nabla \\
\frac{\partial \mu}{\partial q} = \frac{4\pi R_\odot^3}{M_\odot}\rho\zeta^2\frac{\psi}{\theta} \\
\exp(\xi) = \exp(\xi_g) + P_{\rm tur} \\
\gamma = \ln\rho(\exp(\xi_g), \exp(\eta), X_i) \\
\frac{\partial \lambda}{\partial q} = \frac{M_\odot}{L_\odot}\left[\epsilon - \frac{\partial U}{\partial t} + \frac{P}{\rho^2}\frac{\partial \rho}{\partial t}\right]\frac{\psi}{\theta} \\
\frac{\partial \zeta}{\partial q} = \frac{\psi}{\theta} \\
\frac{\partial \psi}{\partial q} = 0 \\
\frac{\partial X_i}{\partial t} = -\frac{2}{3M_\odot\sqrt{\nu}}\frac{\partial F_i}{\partial \mu} + \Psi_i(\xi_t, \eta; \mathcal{X}),\ 1 \le i \le n_{\rm elem}
\end{array}\right.$$

$$\left\{\begin{array}{l}
\text{conditions limites} \\
\mu(n,t) = \mu_{\rm ext}\left(\lambda(n,t)L_\odot, \zeta(n,t)R_\odot\right), \\
\xi(n,t) = \ln\left(P_{\rm ext}\left(\lambda(n,t)L_\odot, \zeta(n,t)R_\odot\right)\right), \\
\eta(n,t) = \ln\left(T_{\rm ext}\left(\lambda(n,t)L_\odot, \zeta(n,t)R_\odot\right)\right), \\
\xi_{\rm g}(n,t) = \xi(n,t),\ \gamma(n,t) = \ln\rho(P_{\rm g}(n,t), T(n,t), X_i), \\
\zeta(1,t) = 0,\ \lambda(1,t) = 0,\ \mu(1,t) = 0. \\
\text{conditions initiales} \\
X_i(q,0) = X_{i,0},\ i = 1, \ldots, n_{\rm elem}, \\
\xi(q,0) = \xi_0(q),\ \eta(q,0) = \eta_0(q),\ \gamma(q,0) = \gamma_0(q), \\
\zeta(q,0) = \zeta_0(q),\ \lambda(q,0) = \lambda_0(q), \\
1 \le q \le n.
\end{array}\right.$$

On a utilisé la notation : $\nu \equiv \mu^{2/3}$. Ces équations de l'espace physique sont formées dans la routine `static_r`, *cf.* § 8.123 (Page 204). Ainsi, la condition de limite libre est, en fait, déjà incluse dans les équations écrites avec la fonction de répartition et n'entraine aucune modification fondamentale de structure.

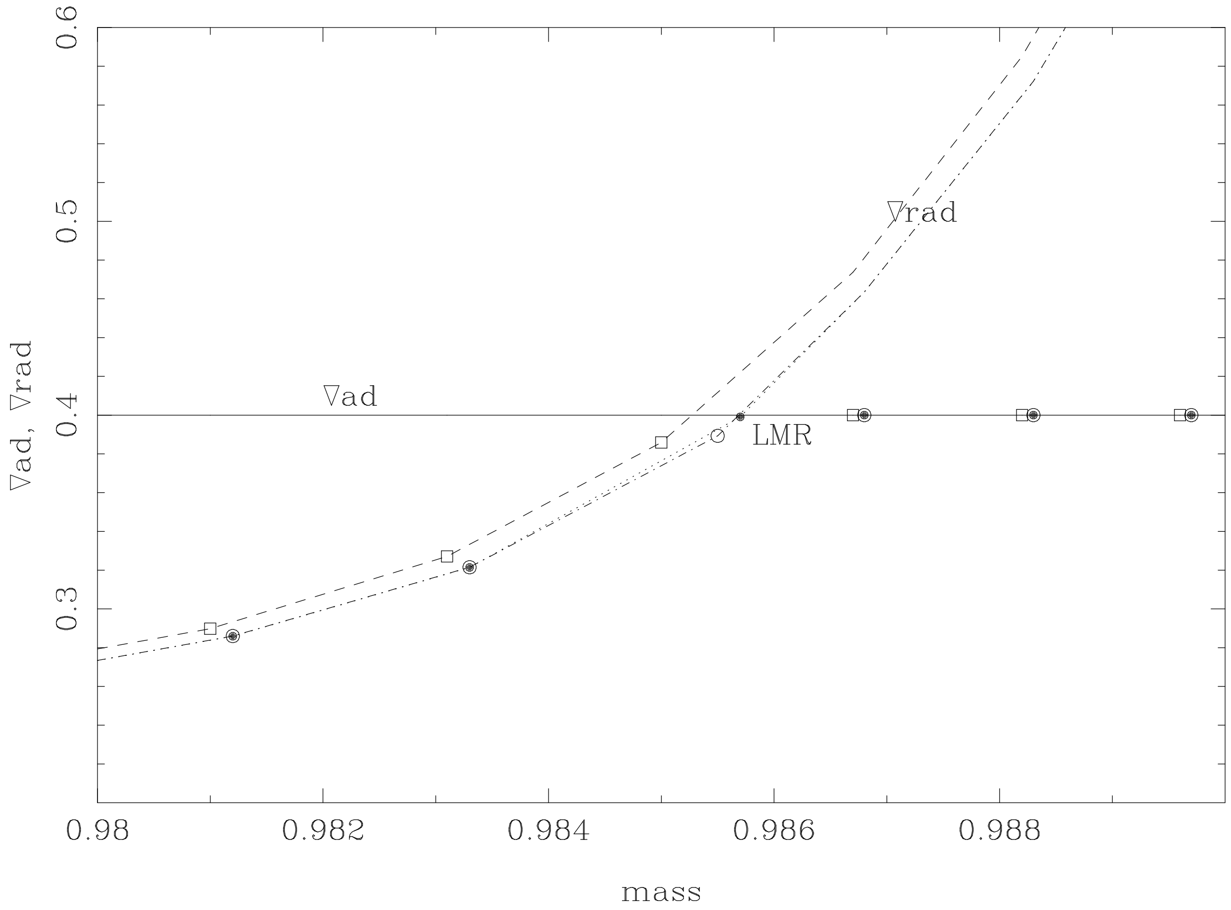

Fig. 7.3 – Deux itérations ont été nécessaires pour positionner un point de grille sur une limite zone radiative / zone convective. Au cours du processus itératif, le gradient radiatif $\nabla_{\rm rad}$ a légèrement diminué, et la limite s'est déplacée vers la droite, alors que le gradient adiabatique $\nabla_{\rm ad}$ restait inchangé. Les points de grille se sont successivement déplacés de leurs positions initiales, carrés vides, vers leurs positions finales, cercles vides puis cercles pleins.

7.2.5 Nœud sur une limite zone radiative / zone convective

A chaque limite entre une zone convective et une zone radiative le gradient :

$$\nabla \equiv \left(\frac{\partial \ln T}{\partial \ln P}\right)_t$$

n'est pas dérivable. Il en est donc de même de la dérivée première de la température. Entre deux points de la grille d'intégration, les inconnues sont représentées par des polynômes par morceaux. Une limite entre une zone radiative et une zone convective doit donc nécessairement se trouver sur une maille du réseau. Avec CESAM2k il suffit de placer un point de grille à mieux que $\sim 5\%$ de la distance entre les nœuds entourant immédiatement chaque limite entre une zone convective et une zone radiative, il n'y alors pas de risque d'approcher les fonctions inconnues par un morceau de polynôme traversant une discontinuité, les points de collocation étant situés à une distance des points de grille, supérieure à 5% de la largeur de la maille concernée.

Pour assurer cette disposition, on utilise un paramètre libre de la fonction de répartition. D'une maille à l'autre, la variation de la fonction de répartition $Q(\mu, t)$, est constante. Les facteurs de répartition a et b définissent les poids relatifs des variables à l'intérieur de chaque maille, ces poids sont définis à un facteur multiplicatif près. Jusqu'à présent ce paramètre libre a été, implicitement, posé égal à l'unité. La modification de cette valeur pour les mailles situées au voisinage d'une limite entre une zone radiative et une zone convective va permettre de rejeter la limite dans un voisinage immédiat des points de grille. On définit la fonction en escalier $\omega(q)$, ajustable d'une maille à l'autre par :

$$Q(\mu, t) = \omega(q) Q_0(\mu, t) \tag{7.7}$$

$Q_0(\mu, t)$ notant maintenant la fonction de répartition retenue, c'est à dire celle définie par la relation Eq. 7.6 (Page 71). Tout en respectant la constante du saut de $Q(\mu, t)$ d'une maille à l'autre, $\omega(q)$ permet d'ajuster le saut de Q_0 dans chaque maille, donc les sauts de ξ, η, ζ, λ et μ de façon à ce qu'ils correspondent à la quantité nécessaire pour atteindre exactement la limite. Une fois la condition :

$$\frac{\partial^2 Q}{\partial q^2} = \frac{\partial \psi}{\partial q} = 0$$

satisfaite en tous points du réseau, dans une maille donnée, la variation de la fonction Q étant constante :

$$\frac{\partial Q}{\partial q} = \psi = cte = \frac{\partial \omega Q_0}{\partial q} = \omega \frac{\partial Q_0}{\partial q} \Rightarrow \omega = \psi / \frac{\partial Q_0}{\partial q} \sim \psi / \frac{\Delta Q_0}{\Delta q} = \psi / \Delta Q_0$$

puisque d'une maille à la suivante[6] $\Delta q = 1$. Pour chaque maille $[i, i+1], i = 1, \ldots, n-1$, on détermine ω_i de façon à ce que le saut de Q_0 :

$$\Delta Q_0 = a(\xi_{i+1} - \xi_i) + b(\mu_{i+1} - \mu_i)$$

corresponde à ce qu'il est nécessaire, pour que chaque limite entre une zone convective et une zone radiative se place sur un point du réseau[7].

Malheureusement il n'y a pas de relation permettant de prévoir le déplacement d'une limite entre une zone convective et une zone radiative en fonction des variations des variables locales, si bien qu'il est nécessaire de procéder par ajustements successifs, il en résulte que la convergence du processus itératif global ne peut être que du premier ordre.

A chaque itération de la méthode de Newton-Raphson, la limite entre une zone radiative et une zone convective est localisée par interpolation linéaire en fonction de la variable d'indice q. Suivant la position interpolée pour les limites, les mailles concernées sont agrandies ou resserrées. A l'aide de la valeur obtenue pour la constante $\psi(t)$ et des valeurs de la variable Q_0, de nouvelles valeurs de $\omega(q)$ sont calculées pour chaque maille. Un exemple d'un tel ajustement est illustré Fig. 7.3 (Page 73).

[6] C'est cette dernière relation qui définit $\omega(q)$, une fonction en escalier n'étant pas dérivable ! Pour faciliter l'exposé on a préféré introduire $w(q)$ par Eq. 7.7 (Page 74).

[7] Des relations similaires peuvent être obtenues avec les facteurs de répartition $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$, $\Delta\lambda$.

Dans la routine `lim_zc`, *cf.* § 8.69 (Page 174), les limites zones radiatives / zones convectives sont localisées avec précision, ce qui permet de déterminer les poids à affecter à chaque maille ; l'expérience des calculs montre que le positionnement de la limite s'effectue sur un nœud avec une précision supérieure à 1%, si la limite est bien définie. Quand la transition zone radiative / zone convective est floue *i.e.* quand les gradients adiabatique et radiatif sont proches au voisinage de la limite, le positionnement n'est pas précis *cf.* § 6.2 (Page 52). Dans ces cas, la discontinuité de la dérivée du gradient est *petite* et l'erreur qui en résulte n'est importante ni pour les variables de structure ni pour la composition chimique qui ne peut, avec une limite floue, que présenter une discontimuité peu marquée.

7.3 Restitution de l'atmosphère

La lumière parvenant des étoiles est la principale source d'informations permettant d'induire leur structure. Un modèle d'évolution stellaire doit décrire le transfert du rayonnement à travers l'atmosphère. Bien que cette dernière ne représente qu'une infime partie de l'étoile, le calcul de ce transfert est un problème complexe qui ne peut être raisonnablement résolu simultanément avec celui de la structure interne. Une des difficultés réside dans le fait que c'est dans l'atmosphère que le rayonnement transite du milieu opaque de l'intérieur où il est quasi isotrope, vers le milieu interstellaire où il est anisotrope. La nécessité est de transporter les conditions limites du milieu transparent, où elles sont définies par l'observation, vers le milieu épais de l'intérieur. La restitution de l'atmosphère consiste alors à construire un modèle aussi précis que possible, assurant le transfert de l'énergie lumineuse et l'équilibre quasi-statique.

Les conditions limites externes du problème différentiel de la structure interne portent sur deux des trois variables thermodynamiques P, T et ρ. CESAM$2k$ offre la possibilité d'utiliser deux types de méthodes : l'approximation monocouche et la reconstitution de l'atmosphère, cette dernière constituant un problème différentiel aux limites résolu numériquement par collocation, *cf.* Morel *et al.* [1994].

7.3.1 L'approximation monocouche

Le "Solar Model Comparison Project", J. Christensen–Dalsgaard [1988], utilise une atmosphère simplifiée pour transporter les conditions limites externes dans le milieu optiquement épais ; elle est obtenue à l'aide de deux relations déduites d'une discrétisation simple des équations de l'atmosphère :

$$\begin{cases} \frac{dP}{d\tau} = \frac{GM_\star}{R_\star^2\kappa} - \frac{2\Omega^2 R_\star}{3\kappa}, \ \frac{\Delta P}{\Delta\tau} \sim \frac{P-P_{\rm ext}}{\tau-0} \\ L_\star = 4\pi R_\star^2 \sigma T^4 \end{cases} \Rightarrow \begin{cases} P = \frac{GM_\star}{R_\star^2\kappa} - \frac{2\Omega^2 R_\star}{3\kappa} \\ L_\star = 4\pi R_\star^2 \sigma T^4 \end{cases}$$

sur la couche externe on a alors $T{\rm ext} = T_{\rm eff}$; σ est la constante de Stéfan, $R_\star$, $M_\star$, Ω et $L_\star$ sont respectivement, le rayon, la masse, la vitesse angulaire et la luminosité de l'étoile. Les deux dernières relations étant obtenues avec $\tau = 1$ et $P_{\rm ext} = 0$ ce qui, physiquement, peut être interprété en disant que les conditions limites externes sont écrites à $\tau = 1$. Cette formulation est utilisée dans la routine `lim_tau1`, *cf.* § 8.68 (Page 174). L'évolution est arrêtée si l'accélération centrifuge devient supérieure à

90% de la la gravité – ce qui se traduirait par une pression négative.

Dans la routine `lim_gong1`, *cf.* § 8.68 (Page 174), deux facteurs $\beta = 7.22$ et $\lambda = 6$, affectent respectivement les équations de la pression et de la luminosité, leur effet étant, avec une équation d'état supposant l'ionisation totale, d'obtenir un modèle solaire qui ressemble au soleil actuel.

PB `Avec l'approximation monocouche, la pression turbulente est ignorée dans l'atmosphère.`

7.3.2 Reconstitution d'une atmosphère

Dans les cas où une grande précision doit être atteinte, en particulier pour les modèles solaires servant au calcul des oscillations, il n'est pas possible d'ignorer la structure fine des couches externes où se réfléchissent les modes d'oscillation. Il est nécessaire de reconstituer le moins mal possible cette couche limite ; c'est l'objet de la routine `lim_atm`, *cf.* § 8.67 (Page 173). Dans un calcul de structure interne, il n'est pas réaliste de traiter le transfert radiatif avec toute sa complexité. L'approximation de diffusion, cas gris, utilisé dans la structure interne, n'est justifiée que pour des profondeurs optiques Rosseland $\tau_{\rm rac} \gtrapprox 20$, limite au delà de laquelle, *cf.* Morel *et al.* [1994], le flux radiatif calculé par l'approximation de diffusion et le flux réel ne diffèrent que par moins de 1%. Pour éviter de traiter explicitement l'atmosphère, on transporte de $\tau \sim 0$ à $\tau = \tau_{\rm rac}$ les conditions externes *i.e.* celles au raccord avec le milieu interstellaire, à savoir : la densité ou la pression, la température et la masse. Avec CESAM2*k*, afin de relier le modèle de l'atmosphère et celui de la structure interne, il faut disposer pour la pression, la température et la masse de trois fonctions[8] au point de raccord en $\tau = \tau_{\rm rac}$ *i.e.* à la limite externe de l'enveloppe :

$$P_{\rm rac}(R, L),\ T_{\rm rac}(R, L),\ M_{\rm rac}(R, L),$$

avec $M_{\rm rac}(R, L) \simeq M_{\rm tot}$ masse contenue à l'intérieur de la couche la plus externe de l'enveloppe. Dans l'atmosphère, la luminosité L est supposée constante $L = L_\star$. La composition chimique, la vitesse angulaire qui, éventuellement, varient en fonction du temps *e.g.* par diffusion et mélange convectif, sont supposées constantes et fixées à leur valeurs sur la couche la plus extérieure de l'enveloppe. Pour une longueur de mélange et une composition chimique fixées, la température est donnée dans l'atmosphère par une loi $T(\tau)$ de la forme : $T(\tau_{\rm R}, T_{\rm eff}, g)$ dépendant, de la profondeur[9] optique Rosseland $\tau_{\rm R}$, $d\tau = -\kappa\rho dR$, de la température effective $T_{\rm eff}$ et de la gravité g :

$$T_{\rm eff} = \left(\frac{L}{4\pi R_{\rm tot}^2 \sigma}\right)^{\frac14}, \quad g = \frac{GM}{R^2} \cong \frac{GM_{\rm tot}}{R_{\rm tot}^2},$$

$R_{\rm tot}$ est le rayon au point de raccord avec le milieu interstellaire *cf.* Mihalas, [1978] p. 49, et κ est l'opacité.

[8]On assurera ensuite, par itérations, l'égalisation de ces valeurs résultant de l'intégration de l'atmosphère et de celles résultant de l'intégration de la structure interne.

[9]Pour plus de lisibilité l'indice R pour "Rosseland" est omis dans la suite.

Cette définition de la température effective correspond à la température du corps noir rayonnant la même densité d'énergie que l'étoile, elle n'est pas universelle. Une seconde définition, souvent utilisée, permet de définir un "rayon observable" $R_\star$:

$$R_\star = R(\tau = \frac{2}{3}) \text{ et } T_{\rm eff} = \left(\frac{L}{4\pi\sigma R_\star^2}\right)^{\frac{1}{4}}.$$

Dans ces conditions, la masse $M_\star < M_{\rm tot}$ est celle à la profondeur optique $\tau = 2/3$: $M_\star = M(\tau = 2/3)$.
Dans CESAM2k, une troisième définition de $T_{\rm eff}$, *cf.* Schatzman & Praderie [1990], est utilisée : le rayon de l'étoile $R_\star$ est défini à la profondeur optique à laquelle la température est égale à la température effective, $R_\star$ est le "rayon bolométrique". Pour certaines lois $T(\tau)$, les deux dernières définitions sont équivalentes. Par la suite, on notera $\tau_\star$ la profondeur optique où le rayon est celui de l'étoile : $R(\tau_\star) = R_\star$.
La composition chimique, l'opacité et la définition retenue pour la température effective doivent, évidemment, être les mêmes dans :

- le programme de calcul du modèle d'atmosphère à partir duquel est déduite la loi $T(\tau, T_{\rm eff}, g)$,
- la partie atmosphère du programme de structure interne,
- la partie supérieure de l'enveloppe.

Connaissant $R_{\rm rac}$ et L on peut intégrer de $\tau = \tau_{\rm min}$ à $\tau = \tau_{\rm rac}$ où on obtient, au rayon $R_{\rm rac}$, point de raccord entre l'enveloppe et l'atmosphère, les trois quantités :

$$P_{\rm rac}(R, L),\ T_{\rm rac}(R, L),\ M_{\rm rac}(R, L).$$

En utilisant τ comme variable indépendante, le système des équations à intégrer est :

$$\left\{\begin{array}{l} \frac{dP}{d\tau} = \frac{GM}{\kappa R^2} - \frac{2\Omega^2 R}{3\kappa} \\ \frac{dT}{d\tau} = \frac{T}{P}\frac{dP}{d\tau}\nabla \\ \frac{dR}{d\tau} = -1\,/\,\kappa\rho \\ \frac{dM}{d\tau} = -4\pi R^2\,/\,\kappa \\ \frac{dR_\star}{d\tau} = 0 \\ \frac{d\tau_\star}{d\tau} = 0 \\ P_{\rm gaz} = P\left\{1 - \frac{a\alpha^2\delta}{8}\frac{\Gamma}{\Gamma+1}(\nabla - \nabla^*_{\rm ad})\right\} \end{array}\right. \qquad \left\{\begin{array}{l} \text{conditions limites} \\ \rho(\tau_{\rm min}) = \rho_{\rm ext} \\ \text{ou } P(\tau_{\rm min}) = g\kappa\tau_{\rm min} \\ P_{\rm gaz}(\tau_{\rm min}) = P(\tau_{\rm min}) \\ M(\tau_\star) = M_\star \\ R(\tau_\star) = R_\star \\ T_{\rm eff} = \mathcal{T}(\tau_\star, T_{\rm eff}, g_\star) \\ T(\tau_\star) = \mathcal{T}(\tau_\star, T_{\rm eff}, g_\star) \\ R(\tau_{\rm rac}) = R_{\rm rac} \end{array}\right. \tag{7.8}$$

Les deux équations différentielles et les deux conditions limites portant sur $R_\star$ et $\tau_\star$ ont été ajoutées pour résoudre le problème différentiel avec la limite libre à $\tau = \tau_\star$ *cf.* Stoer & Bulirsch [1979] §7.3.0, elles couplent $R_\star$ et $\tau_\star$ aux autres variables. En codant `lim_ro=.TRUE.` dans le fichier de données, à la limite optiquement mince, définie par $\tau = \tau_{\rm min}$, la condition limite porte sur la densité fixée à la valeur $\rho_{\rm ext}$ donnée par le modèle d'atmosphère ; avec `lim_ro=.FALSE.`, la condition limite porte sur la pression ; on utilise alors la relation dérivée de l'approximation monocouche. La condition sur la pression permet de s'affranchir de la contrainte sur la densité à une profondeur optique fixée arbitrairement. En général on fixe $\tau_{\rm min} \equiv 10^{-4}$ ce

qui correspond, approximativement, au minimum de température dans la photosphère solaire, celà ne constitue évidemment pas une règle absolue[10]. A ce niveau, l'atmosphère est supposée radiative si bien que $P_{\rm gaz} \equiv P$.
La dernière équation qui relie P et $P_{\rm gaz}$ est celle qui est vérifiée dans les zones radiatives où l'efficacité de la convection est nulle *i.e.* $\Gamma \equiv 0$.
Les conditions limites sont donc écrites aux trois niveaux : $\tau = \tau_{\rm rac}$, $\tau = \tau_\star(t)$ et $\tau = \tau_{\rm min}$, en se rappelant que $\tau_\star(t)$ est susceptible de varier au cours du temps. De plus, $\rho_{\rm ext}$ dépend aussi du temps, puisque dépendant de la gravité $\rho_{\rm ext}(g_\star(t))$.
Au temps t, la solution des équations Eq. 7.8 (Page 77) donne $P_{\rm rac}$, $T_{\rm rac}$ et $M_{\rm rac}$ pour R et L donnés.
Evidemment, autant que faire se peut, la loi $T(\tau)$ et la structure interne doivent être calculées avec des physiques aussi équivalentes que possible à savoir : équation d'état, opacité, longueur de mélange, composition chimique *etc...* Malgré de nombreuses tentatives, à cause des incohérences entre la physique, les données, les paramètres, utilisés dans le modèle d'atmosphère d'une part, et dans la structure interne d'autre part, il s'est avéré impossible de raccorder non seulement les fonctions (*i.e.* pression, température, masse), mais aussi leurs gradients à la limite entre atmosphère et enveloppe ; si le gradient de pression *i.e.* $\frac{dP}{dr} = -g\rho$, se raccorde parfaitement, il n'en est pas de même du gradient de température ni, parconséquent, du gradient $\nabla \equiv \frac{d\ln T}{d\ln P}$; des différences de l'ordre de 0.1 à 0.2 pouvant exister de part et d'autre du raccord. Pour certaines applications de telles différences sont prohibitives et, au risque de ne pas vérifier parfaitement la loi $T(\tau)$, un artifice numérique semi-original a été mis au point pour assurer aussi le raccord de ∇.

PB `La restitution de l'atmosphère n'est effectuée que si l'accélération centrifuge est inférieure à 90% de la gravité ; si tel n'est pas le cas, l'évolution est arrêtée.`

7.3.3 Artifices numériques utilisés pour le raccord de ∇

La gravité g étant fixée, la loi $T(\tau)$ est supposée pouvoir s'écrire sous la forme : $T^4 = \frac{3}{4}T_{\rm eff}^4 f(\tau)$; en utilisant l'expression de la luminosité, $L = 4\pi R_\star^2 \sigma T_{\rm eff}^4$, il vient :

$$T^4 = \frac{3}{4\pi ac}\frac{L_\star}{R_\star^2}f(\tau), \quad (\sigma = \frac{ac}{4}), \text{ d'où } \frac{dT}{d\tau} = \frac{3}{16\pi ac}\frac{L_\star}{R_\star^2 T^3}\frac{df}{d\tau},$$

si bien qu'avec $d\tau = -\kappa\rho dr$:

$$\begin{aligned}\nabla \equiv \frac{d\ln T}{d\ln P} &= -\frac{P}{T}\frac{dT}{d\tau}\kappa\rho\frac{dr}{dP} = \frac{P\kappa}{T}\frac{dT}{d\tau}\frac{R^2}{GM} = \frac{3}{16\pi acG}\frac{P\kappa L_\star}{MT^4}\left(\frac{R}{R_\star}\right)^2\frac{df}{d\tau} = \\ &= \nabla_{\rm rad}\frac{L}{L_\star}\left(\frac{R}{R_\star}\right)^2\frac{df}{d\tau} \qquad (7.9)\end{aligned}$$

puisque le gradient radiatif a pour expression :

$$\nabla_{\rm rad} = \frac{3}{16\pi acG}\frac{P\kappa L}{MT^4}.$$

[10]Il faut s'assurer que l'équation d'état et les tables d'opacité donnent des valeurs réalistes pour les conditions physiques auxquelles on peut s'attendre jusqu'à $\tau_{\rm min}$.

Ici $L \equiv L_\star$ et, dans le cas solaire $R/R_\star < 1 \pm 10^{-3}$. Dans une zone convective le flux total F est la somme du flux radiatif et du flux convectif : $F = F_{\rm rad} + F_{\rm conv}$. *Si et seulement si* l'approximation de diffusion est vérifiée *i.e.* à la limite $\tau \gg 1$, le flux radiatif $F_{\rm rad}(\tau) = \int_0^\infty F_\nu d\nu$ a pour expression :

$$F_{\rm rad} = \frac{4acT^4}{3\kappa\rho H_{\rm p}}\nabla$$

avec pour le flux total F :

$$F = \frac{4acT^4}{3\kappa\rho H_{\rm p}}\nabla_{\rm rad}$$

alors :

$$\lim_{\tau\gg1}\frac{F_{\rm rad}}{F} = \left(\frac{R}{R_\star}\right)^2\frac{df}{d\tau} \sim \frac{df}{d\tau},$$

puisque dans l'atmosphère, du moins pour les étoiles naines, le quotient $(R/R_\star)^2 \sim 1 \pm 10^{-8}$ peut être pris égal à l'unité.

L'artifice numérique utilisé pour le raccord de ∇ repose sur la formulation Eq. 7.9 (Page 78) : pour le calcul de la température, au lieu d'utiliser explicitement la loi $T(\tau)$, on intègre l'équation différentielle :

$$\frac{dT}{d\tau} = \frac{T}{P}\frac{dP}{d\tau}\nabla = \frac{T}{P}\frac{dP}{d\tau}\nabla_{\rm rad}\left(\frac{R}{R_\star}\right)^2\frac{df}{d\tau},$$

et une condition limite sur la température, cette dernière étant écrite, *par exemple*, en $\tau_\star$:

$$T(\tau_\star) = T(\tau_\star, T_{\rm eff}, g_\star).$$

Ainsi, à un changement de variable près, les équations *sont les mêmes* dans l'atmosphère restituée et dans la structure interne.

Pour des lois $T(\tau)$ purement radiatives, telles que celles d'Eddington ou de Hopf, $\lim_{\tau\gg1}\frac{df}{d\tau} = 1$. On remarque qu'alors, le flux total est posé égal au flux radiatif même si l'approximation de diffusion est injustifiée. Pour des lois $T(\tau)$ incorporant la convection, telles que celles dérivées de modèles d'atmosphères solaires, $\lim_{\tau\gg1}\frac{df}{d\tau} < 1$ (typiquement $\frac{df}{d\tau} \sim 0.2$ à $\tau = 20$). Il faudra donc un traitement différent suivant que la loi $T(\tau)$ est, ou non, purement radiative.

7.3.4 Loi $T(\tau)$ purement radiative

La loi $T(\tau)$ ne contient pas la partie convective de l'atmosphère, force est donc d'utiliser le formalisme de la structure interne pour le calcul du gradient dans la partie convective, sachant qu'elle sera incorrecte tant que l'approximation de diffusion ne sera pas justifiée *i.e.* tant que $\tau \lessapprox 20$. Dans la partie radiative le gradient aura l'expression Eq. 7.9 (Page 78), dans la partie convective, son expression sera celle déduite de la théorie de la convection utilisée *cf.* § 7.10.2 (Page 127),

la continuité du gradient sera assurée s'il est calculé avec des expressions qui ont mêmes limites de part et d'autre de la transition convectif/radiatif, ce qui est réalisé en utilisant, dans le formalisme de la longueur de mélange, un $\nabla_{\rm rad}$ *modifié* :

$$\nabla^*_{\rm rad} \equiv \nabla_{\rm rad}\frac{df}{d\tau}.$$

Cet artifice numérique est dû à Henyey *et al.* [1965], M. Gabriel & J. Christensen-Dalsgaard, *communications privées*. A la limite $\tau \gg 1$, le gradient sera ainsi exactement celui donné par le formalisme de la longueur de mélange puisque, pour une loi purement radiative, $\lim_{\tau\gg1} \frac{df}{d\tau} = 1$.

Lois $T(\tau]$ purement radiatives implantées dans CESAM2*k* :

- `hopf` : Loi $T(\tau)$ de Hopf, *cf.* § 8.59 (Page 170).
- `edding` : Loi $T(\tau)$ d'Eddington, *cf.* § 8.44 (Page 162).

7.3.5 Lois $T(\tau)$ avec convection

L'artifice numérique précédent n'est pas utilisable puisque $\lim_{\tau\gg1} \frac{df}{d\tau} < 1$ et, par ailleurs, l'utilisation d'une loi $T(\tau)$ non purement radiative perdrait de son intérêt, puisque la description de la partie convective aux profondeurs optiques $\tau \sim 1$ serait effectuée avec une théorie de la convection qui utilise une expression du flux radiatif déduite de l'approximation de diffusion. Dans ce cas, pour assurer la continuité du gradient, l'artifice numérique utilisé consiste en une interpolation linéaire, en fonction de l'épaisseur optique, entre la valeur du gradient déduit de la loi $T(\tau)$:

$$\nabla = \nabla_{\rm rad} \left(\frac{R}{R_\star}\right)^2 \frac{df}{d\tau} \tag{7.10}$$

et de celle $\nabla_{\rm mel}$ calculée par la théorie de la longueur de mélange ainsi qu'il est décrit § 7.10.2 (Page 127). Ainsi, pour les profondeurs optiques $\tau \lessapprox 1$ le gradient est proche de celui dérivé de la loi $T(\tau)$, tandis qu'il tend vers celui de la longueur de mélange pour les profondeurs optiques $\tau \gtrapprox \tau_{\rm rac}$. L'interpolation linéaire utilisée a pour expression :

$$\nabla = (1-x)\nabla_{\rm rad} \left(\frac{R}{R_\star}\right)^2 \frac{df}{d\tau} + x\nabla_{\rm mel},\ x = \frac{\tau - 1}{\tau_{\rm rac} - 1}. \tag{7.11}$$

Pour les profondeurs optiques intermédiaires $\tau \sim 1$, cet artifice numérique permet de restituer, en partie, le modèle d'atmosphère avec convection, tout en assurant, plus profond, le raccord des gradients malgré les incohérences de la physique.

$\boxed{\textbf{PB}}$ `Avec une loi` $T(\tau)$ `non purement radiative, la valeur` $\tau_{\rm rac} = 20$ `est imposée indépendamment de celle indiquée pour` $\tau_{\rm max}$ `dans le fichier de données.`

Lois $T(\tau]$ non purement radiatives implantées dans CESAM2*k* :

- `k5750` : Loi $T(\tau)$ solaire $T_{\rm eff} = 5750$ K calculée par C. Van't Veer avec Atlas 9 de Kurucz, *cf.* § 8.65 (Page 172).
- `k5777` : Loi $T(\tau)$ solaire $T_{\rm eff} = 5777$ K calculée par C. Van't Veer avec Atlas 12 de Kurucz, *cf.* § 8.65 (Page 172).
- `roger00` : Lois $T(\tau)$ à [Fe/H]=0, calculs avec Atlas 12 de Kurucz, interpolations de R. Cayrel. Utilise la table `fesh00.data` du directory `DATA` de la source de CESAM2*k*; avant utilisation indiquer, dans le sous programme, le chemin d'accès de la table, *cf.* § 8.118 (Page 201).
- `roger02` : Lois $T(\tau)$ à [Fe/H]=-0.2, calculs avec Atlas 12 de Kurucz, interpolations de R. Cayrel. Utilise la table `fesh02.data` du directory `DATA` de la source de CESAM2*k*; avant utilisation indiquer, dans le sous programme, le chemin d'accès de la table, *cf.* § 8.118 (Page 201).

- `roger05` : Lois $T(\tau)$ à [Fe/H]=-0.5, calculs avec Atlas 12 de Kurucz, interpolations de R. Cayrel, *cf.* §8.118 (Page 201).
- `roger10a` : Lois $T(\tau)$ à [Fe/H]=-1.0 enhanced, calculs avec Atlas 12 de Kurucz, interpolations de R. Cayrel, *cf.* §8.118 (Page 201).

7.3.6 Résolution numérique

Pour résoudre numériquement le problème différentiel défini par le système Eq. 7.8 (Page 77), on impose (dans la routine `lim_atm`) un indice *fixe* $n_\star = 3/4 n_\mathrm{a}$ à la limite libre $\tau = \tau_\star(t)$ parmi les n_a points de grille que comporte l'atmosphère. Pour ce faire, on utilise la **bijection linéaire par morceaux** : $\tau \mapsto \omega$ de $[\tau_\mathrm{rac}, \tau_\mathrm{min}]$ dans $[1, n_\mathrm{a}]$ définie par :

$$\ln\tau \mapsto \varphi(\omega) = \begin{cases} \ln\tau_\mathrm{rac} + (\omega - 1)\Delta\varphi^+, & \text{si } \omega \in [1, n_\star]\,; \\ \ln\tau_\mathrm{min} + (\omega - n_\mathrm{a})\Delta\varphi^-, & \text{sinon.} \end{cases} \tag{7.12}$$

Les deux valeurs de la pente $\Delta\varphi^+$ et $\Delta\varphi^-$ étant définies par :

$$\Delta\varphi^+ \equiv \frac{\ln\tau_\star - \ln\tau_\mathrm{min}}{n_\star - n_\mathrm{a}},\ \Delta\varphi^- \equiv \frac{\ln\tau_\star - \ln\tau_\mathrm{rac}}{n_\star - 1};$$

ainsi on a :

$\omega = 1$, au fond de l'atmosphère où $\tau = \tau_\mathrm{rac}$,

$\omega = n_\star$, sur la limite libre interne à $\tau = \tau_\star$,

$\omega = n_\mathrm{a}$, sur la limite externe où $\tau = \tau_\mathrm{min}$.

On travaille donc avec un pas de grille constant, ce qui est un avantage de la méthode. Pour l'intégration numérique, on a utilisé les variables : $\xi = \ln P$, $\xi_\mathrm{g} = \ln P_\mathrm{gaz}$, $\eta = \ln T$, $\zeta_\mathrm{a} = R/R_\odot$, $\mu_\mathrm{a} = M/M_\odot$, $\zeta_\star = R_\star/R_\odot$ et $\varphi_\star = \ln\tau_\star$, si bien que le système Eq. 7.8 (Page 77) devient :

$$\begin{cases} \frac{d\xi}{d\omega} = \left[\frac{GM_\odot}{R_\odot^2}\frac{\mu_\mathrm{a}}{\zeta_\mathrm{a}^2} - \frac{2R_\odot\Omega^2\zeta_\mathrm{a}}{3}\right]\frac{\Delta\varphi^\pm}{\kappa}\exp(\varphi - \xi) \\ \frac{d\eta}{d\omega} = \frac{d\xi}{d\omega}\nabla \\ \frac{d\zeta_\mathrm{a}}{d\omega} = -\frac{1}{R_\odot}\frac{\Delta\varphi^\pm\exp(\varphi)}{\kappa\rho} \\ \frac{d\zeta_\star}{d\omega} = 0 \\ \frac{d\mu_\mathrm{a}}{d\omega} = -\frac{4\pi R_\odot^2}{M_\odot}\frac{\Delta\varphi^\pm}{\kappa}\exp(\varphi)\zeta_\mathrm{a}^2 \\ \frac{d\varphi_\star}{d\omega} = 0 \\ \frac{d\varphi}{d\omega} = \Delta\varphi^\pm \\ \exp(\xi)\left\{1 - \frac{a\alpha^2\delta}{8}\frac{\gamma}{\gamma+1}(\nabla - \nabla^*_\mathrm{ad})\right\} - \exp(\xi_\mathrm{g}) = 0 \end{cases} \tag{7.13}$$

avec $\Delta\varphi^\pm = \Delta\varphi^+$ (*respt.* $\Delta\varphi^-$) si $\omega \in [1, n_\star]$ (*respt.* $\omega \in [n_\star, n_\mathrm{a}]$).

Les conditions limites sont alors :

$$\begin{cases} \zeta_{\rm a}(1) = \zeta_{\rm rac} \\ \mu_{\rm a}(n_\star) = M_\star/M_\odot \\ \zeta_\star(n_\star) = R_\star/R_\odot \\ \eta(n_\star) = \ln \mathcal{T}(\exp(\varphi_\star), T_{\rm eff}, g_\star) \\ \eta(n_\star) = \ln T_{\rm eff} \\ \varphi(n_{\rm a}) = \ln \tau_{\rm min} \\ \rho(n_{\rm a}) = \rho_{\rm ext} \\ \xi(n_{\rm a}) = \xi_{\rm g}(n_{\rm a}) \end{cases}$$

avec :

$$T_{\rm eff}^4 = \frac{L_\odot\sqrt{\lambda_{\rm rac}^3}}{4\pi R_\odot^2 \sigma \zeta_\star^2}, \quad g = \frac{GM_\odot \mu_{\rm a}}{R_\odot^2 \zeta_{\rm a}^2}.$$

La solution numérique de ce problème différentiel non-linéaire utilise la méthode de spline-collocation. Le processus itératif est initialisé, lors du premier calcul, par un modèle d'atmosphère solaire et, pendant l'évolution, par le modèle d'atmosphère du modèle au pas de temps précédent.

$\boxed{\textbf{PB}}$ `A cause de la discontinuité de la dérivée de la bijection linéaire par morceaux` *cf.* §6.2 (Page 52)`, au point` $\tau = \tau_\star$`, il semble se produire des difficultés de convergence lorsque` $\tau_\star$ `tente de franchir la discontinuité, un grand nombre d'itérations est alors nécessaire. En fait, une propriété de la méthode de spline-collocation permet d'atteindre effectivement la solution.`

Typiquement, on utilise de $n_{\rm a} = 30$ à $n_{\rm a} = 100$ points de grille pour l'intervalle $[\tau_{\rm rac}, \tau_{\rm min}]$. Avec des B-splines d'ordre 3 il y a de 59 à 199 points d'intégration où les Eqs. (7.13) sont satisfaites, en tenant compte de la superconvergence, l'ordre du schéma est 4.

Ces équations sont résolues dans l'espace des splines par la routine `coll_atm`, *cf.* §8.17 (Page 150), en utilisant les coefficients calculés dans l'espace physique par la routine `eqatm` *cf.* §8.45 (Page 162). La gestion et l'interface entre les deux espaces étant assurée par la routine `lim_atm`, *cf.* §8.67 (Page 173). Avec les lois $T(\tau)$ déduites de modèles d'atmosphère sophistiqués, la profondeur optique $\tau_\star$ varie avec la température effective et la gravité, l'équation implicite $T(\tau_\star) = T_{\rm eff}$ est alors résolue dans la routine `taueff`, *cf.* §8.125 (Page 206).
Dans le programme de résolution, le point d'indice 1 correspond au raccord à $\tau = \tau_{\rm rac}$ et $R = R_{\rm rac}$, le point d'indice $n_{\rm a}$ à $\tau = \tau_{\rm min}$. La densité externe $\rho_{\rm ext}$ à $\tau = \tau_{\rm min}$ dépend de la loi $T(\tau)$ utilisée. La profondeur optique $\tau = \tau_{\rm rac}$ du fond de l'atmosphère est un paramètre externe introduit dans la NAMELIST `NL_ATM` du fichier de données : $\tau_{\rm rac}$ =`tau_max`.

La solution consiste en $\xi_{\rm rac}(\zeta_{\rm rac}, \lambda_{\rm rac})$, $\eta_{\rm rac}(\zeta_{\rm rac}, \lambda_{\rm rac})$, $\mu_{\rm rac}(\zeta_{\rm rac}, \lambda_{\rm rac})$; les dérivées par rapport à $\eta_{\rm rac}$ et $\lambda_{\rm rac}$ sont calculées numériquement.

7.4 Evolution temporelle de l'énergie gravifique

L'énergie graviphique a un rôle prépondérant dans les parties de l'étoile où elle se substitue à l'énergie thermonucléaire, lorsque le fuel nucléaire a disparu, par exemple dans le cœur, après la séquence principale ou encore dans la phase d'Hayashi. C'est cette équation qui, traduisant l'hypothèse d'équilibre quasi-statique, évite un traitement hydrodynamique de l'évolution stellaire. Elle n'est pas justifiée dans toutes les phases de l'évolution.

7.4.1 L'approximation de Kippenhahn

Bien qu'elle soit formellement incorrecte *cf.* Cox & Guili [1968], Strittmatter *et al.* [1970], à une abscisse lagrangienne où la variation temporelle du nombre de particules libres n'est pas nulle *e.g.* dans un cœur convectif, l'approximation de Kippenhahn [1968] s'écrit :

$$\epsilon_{\rm G} \sim T\left(\frac{\partial S}{\partial t}\right)$$

ou S est l'entropie spécifique. Cette approximation permet d'exprimer l'énergie gravifique explicitement en fonction des variations de pression et de température :

$$\epsilon_{\rm G} \sim T\left(\frac{\partial S}{\partial t}\right) = c_{\rm P}\frac{\partial T}{\partial t} - \frac{\delta}{\rho}\frac{\partial P_{\rm gaz}}{\partial t}$$

est d'un emploi facile. Pour les étoiles de masse ($M \geq 1.2M_\odot$) l'extension du cœur convectif varie au cours de l'évolution, créant ainsi une discontinuité de la composition chimique donc de la densité, suivant l'axe temporel ; discontinuité dont il y a lieu de tenir compte dans l'évaluation de la dérivée temporelle de la densité dans Eq. 7.14 (Page 83). L'approximation de Kippenhahn qui n'utilise pas explicitement la dérivée d'une fonction discontinue permet de contourner la difficulté et de faire supporter l'erreur par le numérique.

7.4.2 Discrétisation

La variation d'énergie gravifique :

$$\epsilon_{\rm G} = \left(\frac{\partial U}{\partial t}\right)_\mu + P\left(\frac{\partial V}{\partial t}\right)_\mu$$

où U est l'énergie interne spécifique, $V = 1/\rho$ le volume spécifique, et μ la variable d'espace lagrangienne, est approchée dans CESAM2k par un schéma aux différences finies implicite :

$$\epsilon_{\rm G} \simeq \frac{U^{t+\Delta t} - U^t}{\Delta t} - \frac{P^{t+\Delta t}}{(\rho^{t+\Delta t})^2}\frac{\rho^{t+\Delta t} - \rho^t}{\Delta t} \tag{7.14}$$

où les quantités U^t et ρ^t à l'instant t sont calculées en utilisant l'équation d'état avec des valeurs de la pression et de la température obtenues par interpolation à μ *i.e.* masse constante, de la solution précédente :

$$\mu \hookrightarrow q \hookrightarrow P^t, T^t$$

et, pour la composition chimique X, par une interpolation directe par rapport à μ, d'ordre `mc`, ordre des B-splines d'interpolation de la composition chimique.

Avec l'approximation de Kippenhahn, la formulation aux différences finies totalement implicite est écrite :

$$\epsilon_{\rm G} \simeq c_{\rm P}^{t+\Delta t}\frac{T^{t+\Delta t}-T^t}{\Delta t} - \frac{\delta^{t+\Delta t}}{\rho^{t+\Delta t}}\frac{P_{\rm gaz}^{t+\Delta t}-P_{\rm gaz}^t}{\Delta t} \tag{7.15}$$

Ces équations sont formées dans les routines `static_m` et `static_r` *cf.* § 8.123 (Page 204).

7.4.3 Initialisation

Pour le calcul de la pré-séquence principale, il est nécessaire de disposer d'un modèle en équilibre quasi-statique dont la seule source d'énergie est d'origine gravitationnelle. Le modèle initial est obtenu par une méthode due à Iben [1975] qui a été adaptée à CESAM2k en collaboration avec A.Baglin. Au début de la pré-séquence principale l'étoile est complètement convective donc isentropique sauf, peut-être, dans une partie superadiabatique externe. Dans ces conditions, la composition chimique est spatialement et temporellement ***constante***, et l'équation d'énergie s'écrit :

$$\frac{\partial L}{\partial m} = \epsilon_G = -T\frac{\partial S}{\partial t} = cT$$

où c est la "***constante de contraction***" qui caractérise un modèle en équilibre quasi-statique, totalement convectif et sans réactions thermonucléaires. La donnée de la constante de contraction c, permet de résoudre le système des équations au temps $t+dt$ sans avoir à déterminer explicitement la quantité $T\frac{\partial S}{\partial t}$.
La méthode de Iben consiste à construire un premier modèle (indice 1) avec une première valeur c_1 pour c. Puis un second modèle (indice 2) avec une valeur c_2 voisine – CESAM2k utilise $c_2 = 1.1 \times c_1$. Si on admet que ces deux modèles appartiennent à la même séquence évolutive, l'énergie rayonnée pendant l'intervalle de temps Δt les séparant est alors égale à la variation d'énergie gravitationnelle, si bien que :

$$\frac{L_1+L_2}{2}\Delta t \sim \left(\frac{GM^2}{R_2}-\frac{GM^2}{R_1}\right) \Rightarrow \Delta t \sim 2GM^2\frac{R_1-R_2}{(L_1+L_2)R_1R_2}.$$

On peut alors débuter le calcul de l'évolution en utilisant cette valeur de Δt et le modèle 2 comme modèle initial. Une valeur typique pour la constante est $c = 0.02$, la température centrale du modéle initial avoisine alors 100 000K ; on obtient 500 000K (*respt.* 1 000 000K) avec $c = 0.0005$ (*respt.* $c = 0.00008$). Ces ordres de grandeur dépendent assez peu de la masse du modèle. Au cours du déroulement du calcul CESAM2k demande la valeur de la constante de contraction c à utiliser. A chaque étape de la détermination du modèle initial de pré-séquence principale, il est possible de vérifier les valeurs de la température centrale, de la luminosité et du rayon total de façon à pouvoir modifier c jusqu'à obtenir le modèle désiré. L'initialisation pour le calcul du premier modèle peut être faite, soit à partir d'un modèle de pré-séquence principale calculé avec des paramètres voisins, fichier en binaire d'extension `_B.pms`, ou encore avec l'un des fichiers ASCII de modèle de pré-séquence principale `2d-2.pms, 5d-4.pms, 8d-5.pms` du sous-directory `EXPLOIT`. Une fois le modèle initial de pré-séquence principale obtenu celui-ci est écrit, en vue d'une autre initialisation, dans le fichier binaire `mon_modele_B.pms`. L'énergie graviphique initiale de pré-séquence principale est calculée par la routine `iben`, *cf.* § 8.60 (Page 171).

7.5 Evolution de la composition chimique sans diffusion

L'évolution temporelle de la composition chimique pose des difficultés particulières car, même sur la séquence principale où les termes non linéaires des équations n'ont pas encore de rôle déterminant, les équations d'évolution sont délicates à intégrer. Pour ne pas dégrader l'ordre de précision de la résolution des équations de l'équilibre quasi-statique, il faudrait que le schéma utilisé pour le problème de valeurs initiales soit d'ordre aussi élevé que possible. Même sans diffusion, l'analyse montre qu'il est difficile d'utiliser une formule d'intégration d'ordre supérieur à 2 qui satisfasse aux contraintes suivantes imposées par la physique de l'évolution stellaire :

- Coexistence d'échelles de temps d'évolution très différentes. Par exemple le deutérium a un temps d'évolution caractéristique de l'ordre de l'année, celui de l'hydrogène est de l'ordre d'une centaine de million d'années. Un tel problème différentiel est qualifié de "raide".
- Présence de zones convectives qui mélangent les espèces chimiques. Comme il est souhaitable d'utiliser en tout point du modèle la même formule d'intégration, on ne peut qu'utiliser des schémas à un ou deux pas. Au delà, la logique de l'algorithme gérant le mélange convectif et le mouvement des limites devient très compliqué.
- Conservation de la charge, du nombre de nucléons, la formule d'intégration doit donc conserver toute combinaison linéaire des abondances, on dira par la suite qu'elle doit être "conservative".
- Les abondances sont des quantités positives.

Le problème de l'évolution temporelle de la composition chimique étant un problème différentiel raide, afin de justifier les choix faits dans CESAM2k, on donne quelques indications concernant la résolution numérique de ce type de problème ; pour plus d'informations consulter Hairer & Wanner [1991].

7.5.1 Problème raide

Pour intégrer numériquement un problème différentiel de Cauchy :

$$y'(x) = f(x, y), \text{ sur } [x_0, x_N], \ y(x_0) = y_0$$

on se donne un pas $h > 0$ et, de proche en proche, à partir de la valeur initiale y_0, on obtient la solution numérique en utilisant une relation de la forme : $y_n = y_{n-1} + h\phi$, dans laquelle la fonction d'incrémentation ϕ dépend de façon plus ou moins compliquée de la solution obtenue aux points $x_i = x_0 + ih$, $0 \leq i \leq n$, $n \geq 1$.

Ce faisant, deux notions se dégagent : la *convergence* et la *stabilité*. On dit que la formule est convergente si, en $x \in [x_0, x_N]$ fixé ($x = x_0 + nh$, $h = (x - x_0)/n$, $n \leq N$), la limite pour $h \to 0$ de la solution numérique coïncide avec la solution exacte du problème différentiel, soit :

$$\lim_{n\to\infty}(x_0 + nh) = x_N \Rightarrow \lim_{n\to\infty} y_n = y(x_N)$$

On montre que la plupart des formules classiques sont convergentes. La stabilité est reliée au comportement à l'infini de la solution numérique. Une formule d'intégration est stable si, le pas h étant fixé :

$$\lim_{n\to\infty}(x_0 + nh) = \infty \Rightarrow \lim_{n\to\infty} y_n = y(\infty)$$

La non stabilité est due à l'accumulation des erreurs de troncature. Si celles-ci augmentent moins vite que la solution, la formule est stable, instable sinon.

La stabilité dépend, non seulement de la formule d'intégration, mais aussi du problème différentiel posé. L'étude ne peut en être faite que pour des équations simples. Ainsi est-on obligé d'en restreindre la définition : pour qu'il y ait stabilité, il ne faut pas que les erreurs de troncature croissent exponentiellement à l'infini ; pour cette raison on ramène l'étude de la stabilité à celle de la solution numérique du problème type : $y'(x) = \lambda y(x)$, $\lambda \in C$. h étant fixé, la formule est stable si la solution numérique tend vers la solution nulle lorsque n tend vers l'infini. On dit alors que la formule est A-stable. De plus, on la qualifiera de A–positive si, dans le domaine où elle est A-stable et $\lambda \in C$, avec $y_0 > 0$, lorsque n tend vers l'infini, la solution numérique tend vers la solution nulle en restant positive.

De façon assez générale, les formules implicites sont A–stables dans de grandes régions du plan complexe mais peu d'entre elles sont A–positives.

Avec un problème différentiel raide dans lequel les échelles de temps d'évolution des diverses variables sont très différentes la difficulté est double :

- Obtenir une solution stable pour toutes les variables avec un pas temporel acceptable, c'est à dire sans suivre pour autant la plus petite échelle de temps.
- Intégrer les variables significatives avec une précision suffisante.

Pour l'évolution stellaire, à moins qu'on ne désire suivre en détail l'évolution de l'espèce chimique dont l'échelle de temps caractéristique est la plus petite, il faudra utiliser une formule pour problème raide. Ce faisant, il faut toutefois avoir conscience que toutes les variables ne seront pas intégrées avec la même précision, le rôle du schéma pour problème raide est seulement de prévenir contre la croissance exponentielle des erreurs de troncature affectant les variables ayant les plus petites échelles de temps. Par ailleurs, l'analyse du comportement à l'infini du schéma ne peut être faite que sur le problème type à une dimension. Pour un système aussi complexe que celui de l'évolution temporelle de la composition chimique, la théorie n'est pas faite. En transposant à plusieurs dimensions un résultat établi pour une seule dimension il faut s'attendre à ce que, dans certains cas il puisse y avoir des difficultés. Dans ces cas le seul remède simple est de diminuer le pas temporel.

7.5.2 Résumé des contraintes

Pour l'intégration numérique du système d'équations différentielles de l'évolution temporelle de la composition chimique, il est nécessaire d'utiliser une formule :

– Conservative, *i.e.* conservant toute combinaison linéaire des abondances $\sum_i \alpha_i X_i^t = cte$.
– Stable ou, tout au moins A–stable, *i.e.* que les erreurs de troncature relatives aux variables dont l'échelle de temps d'évolution est la plus courte n'augmentent pas exponentiellement lorsque le pas temporel est grand par rapport à cette échelle de temps.
– Positive.
– Simple et précise.

La première condition est remplie pour la plupart des schémas classiques : Runge–Kutta ou schémas à pas liés, pour lesquels la fonction d'incrémentation dépend linéairement des dérivées temporelles estimées en divers points. La condition de A-stabilité est remplie si on utilise un schéma implicite. Il est souhaitable, par ailleurs, d'utiliser le même schéma d'intégration en tous les points du modèle.

Le fait que les limites des zones convectives se déplacent en fonction du temps et la nécessité du mélange convectif élimine malheureusement une partie de l'intérêt des schémas à pas liés qui possèdent des propriétés intéressantes de stabilité pour les problèmes raides *e.g.* schémas BDF. De même, il ne sera possible d'utiliser des schémas de Runge–Kutta d'ordre supérieur à 1, que si on ne tient pas compte du mélange convectif pour les modèles intermédiaires comme avec la formule de Wagoner [1969].

Lorsque la diffusion microscopique est omise, CESAM2k utilise un schéma d'intégration pour problème raide de type Runge-Kutta implicite de type LobattoIIIC.

7.5.3 Les formules IRK Lobatto IIIC.

Dans une zone radiative, sans diffusion, le système des équations de l'évolution de la composition chimique s'écrit formellement :

$$\frac{\partial x_i}{\partial t} = \Psi(T, \rho, \mathcal{X}), 1 \leq i \leq n_{elem}. \tag{7.16}$$

Dans ce qui suit, on décrit l'utilisation de la formule IRK LobattoIIIC, pour alléger le formalisme, on ne considère qu'une seule espèce chimique. On notera x_0 (*respt.* x) l'abondance de cet élément au temps t (*respt.* $t+dt$). Avec les coefficients de la table Table 7.1 (Page 88) le schéma IRK LobattoIIIC est :

$$\begin{aligned} x_i &= x_0 + dt \sum_{j=1}^{s} a_{ij} \Psi(T_j, \rho_j, x_j), \; i = 1, \ldots, s \\ x &= x_0 + dt \sum_{j=1}^{s} b_j \Psi(T_j, \rho_j, x_j) \end{aligned} \tag{7.17}$$

où T_j, ρ_j et x_j sont, respectivement, la température, la densité, l'abondance aux temps intermédiaires, $t + c_j dt$, $j = 1, \ldots, s$; s est le nombre d'étapes de la formule IRK LobattoIIIC.

Toutefois, cette formulation élémentaire d'un schéma de Runge-Kutta ne donne pas des résultats satisfaisants pour le problème de l'évolution des espèces chimiques,

TAB. 7.1 – Coefficients de Eq. 7.17 (Page 87) pour les formules IRK Lobatto IIIC d'ordre $p = 1, 2, 4$ avec $s = 1, 2, 3$ étapes. On utilise la notation usuelle, les c_i, $i = 1, \ldots, s$ sont éléments de la première colonne, les b_i ceux de la dernière ligne, les autres coefficients sont les a_{ij} de la matrice caractérisant la formule de Runge-Kutta.

$$
\begin{array}{c|c}
1 & 1 \\
\hline
 & 1
\end{array}
\qquad
\begin{array}{c|cc}
0 & \frac{1}{2} & -\frac{1}{2} \\
1 & \frac{1}{2} & \frac{1}{2} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
\qquad
\begin{array}{c|ccc}
0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\
\frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}
$$

des valeurs notablement négatives des abondances pouvant survenir ; la cause est la multiplication des erreurs numériques par les constantes de Lipschitz qui ont des valeurs élevées *e.g.* 10^{18}. Cette difficulté est surmontée en écrivant la première équation Eq. 7.17 (Page 87) pour les écarts $z_i = x_i - x_0$ aux quantités initiales *cf.* Hairer & Wanner, *op. cit.* , §IV.8 :

$$
\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_s \end{pmatrix} = dt\ \mathcal{A} \begin{pmatrix} \Psi(T_1, \rho_1, x_0 + z_1) \\ \Psi(T_2, \rho_2, x_0 + z_2) \\ \vdots \\ \Psi(T_s, \rho_s, x_0 + z_s) \end{pmatrix}. \tag{7.18}
$$

La matrice $\mathcal{A} \equiv (a_{i,j})_{i,j=1}^{s}$ d'une quelconque formule IRK Lobatto IIIC étant inversible, en notant $(d_1, d_2, \ldots, d_s) \equiv (b_1, b_2, \ldots, b_s)\mathcal{A}^{-1}$, la seconde équation Eq. 7.17 (Page 87) devient simplement[11] :

$$
x = x_0 + z_s. \tag{7.19}
$$

7.5.4 Mélange des éléments chimiques sans diffusion

Dans une zone convective, lorsque l'échelle de temps de la convection est petite par rapport à celle des réactions nucléaires, la composition chimique est homogénéisée par les mouvements de convection. Pour l'isotope i, le taux de destruction Ψ_i n'est plus la quantité locale $\Psi_i(T, \rho, \mathcal{X})$ mais une quantité moyennée sur toute la zone convective. Ainsi dans une zone convective Eq. 7.16 (Page 87) doit être remplacée par :

$$
\frac{d\bar{x}_i}{dt} = \frac{\int_{ZC} \Psi_i(\rho, T, \bar{\mathcal{X}}, t)dm}{\int_{ZC} dm} = \frac{\int_{ZC} \Psi_i(\rho, T, \bar{\mathcal{X}}, t)\sqrt{\nu}d\nu}{\int_{ZC} \sqrt{\nu}d\nu},\ \nu = \left(\frac{m}{M_\odot}\right)^{\frac{2}{3}} \tag{7.20}
$$

on remarquera que le taux de destruction local Ψ_i est calculé avec $\bar{\mathcal{X}}$ *i.e.* la valeur moyenne des abondances sur la zone mélangée, ce qui évidemment complique la situation.

[11]Cette particularité résulte du fait que la dernière ligne de A est identique au vecteur b *e.g.* toutes les formules IRK Lobatto IIIC sont construites avec cette propriété.

Comme il est rappelé plus haut, on fait l'hypothèse selon laquelle, dans une zone convective, l'abondance de chaque isotope est constante. Pour reprendre l'exemple d'un seul élément, notons $\bar{x}$ sa valeur dans une zone convective, et à chaque étape temporelle $t + c_j dt,\ j = 1, \dots, s$ de l'intégration Runge-Kutta, notons sa valeur moyenne $\bar{x_j}$. On estime par la formule du point milieu les intégrales de Eq. 7.20 (Page 88) si bien que :

$$\bar{x_j} = \sum_{k\in \mathrm{MZ}} x_{j,k+1/2}\Delta\nu_{k+1/2}, \tag{7.21}$$

où la somme porte sur toutes les couches que comporte la zone convective, $x_{0,k+1/2}$ est l'abondance au point milieu, $\nu_{k+1/2} \equiv (\nu_{k+1} + \nu_k)/2$ et :

$$\Delta\nu_{k+1/2} \equiv \frac{(\nu_{k+1} - \nu_k)\sqrt{\nu_{k+1/2}}}{\sum_{j\in \mathrm{MZ}}(\nu_{j+1} - \nu_j)\sqrt{\nu_{j+1/2}}}. \tag{7.22}$$

L'équation 7.18 devient :

$$\begin{pmatrix} \bar{z}_1 \\ \bar{z}_2 \\ \vdots \\ \bar{z}_s \end{pmatrix} = dt\ \mathcal{A} \begin{pmatrix} \sum_k \Psi(T_{1,k+\frac12}, \rho_{1,k+\frac12}, \bar{x_0} + \bar{z}_1)\Delta\nu_{k+\frac12} \\ \sum_k \Psi(T_{2,k+\frac12}, \rho_{2,k+\frac12}, \bar{x_0} + \bar{z}_2)\Delta\nu_{k+\frac12} \\ \vdots \\ \sum_k \Psi(T_{s,k+\frac12}, \rho_{s,k+\frac12}, \bar{x_0} + \bar{z}_s)\Delta\nu_{k+\frac12} \end{pmatrix} \tag{7.23}$$

où $T_{j,k+\frac12}$ (*respt.* $\rho_{j,k+\frac12}$) est la température (*respt.* la densité) au temps intermédiaire $t + c_j dt$; $\bar{x}$ s'écrit :

$$\bar{x} = \bar{x_0} + \bar{z}_s. \tag{7.24}$$

Formellement, cette formulation est aussi valable en un point d'une zone radiative, les sommes ne portant que sur une seule valeur de l'indice k. Pour simplifier, la valeur de $\bar{x_0}$ est obtenue en faisant porter la sommation Eq. 7.21 (Page 89) sur l'intervalle de masse correspondant à la zone convective au temps $t+dt$ qui peut ne pas *exactement* coïncider avec l'intervalle de masse au temps t. L'imprécision engendrée est faible si la zone convective se déplace peu, par contre, dans le cas d'une apparition ou d'une disparition l'erreur engendrée peut être conséquente. Tenir compte de ces subtilités entrainerait de grandes complications dans les algorithmes, car il faudrait introduire explicitement le déplacement de la zone convective dans la relation Eq. 7.23 (Page 89).

L'extension à plusieurs isotopes ne présente que des difficultés algorithmiques assez techniques. Avec l'approximation des intégrales Eq. 7.21 (Page 89) sur la zone convective à l'aide de la formule du point milieu, la précision numérique n'est que du second ordre ; une formule intégrale de Gauss à deux points permettrait d'atteindre le troisième ordre[12]. Dans le cas solaire, l'erreur reste limitée, car les réactions thermonucléaires sont peu efficaces dans la zone convective ; pour le cas stellaire, les imprécisions de la théorie et des données physiques surpassent largement celles engendrées par un ordre d'intégration peu élevé.

Dans la routine `rk_imps` *cf.* § 8.117 (Page 199), les équations Eq. 7.18 (Page 88) à Eq. 7.24 (Page 89) sont résolues itérativement à l'aide du schéma de Newton-Raphson[13] avec un jacobien recalculé pour chaque itération ; la méthode implicite

[12] Ce qui n'entrainerait qu'une complication supplémentaire d'algorithme.

[13] A cause de la raideur, les constantes de Lipschitz sont grandes et un schéma du type predicteur-correcteur est inefficace.

est initialisée avec des valeurs initiales nulles pour les z_i ; la convergence est supposée atteinte dès que les corrections sur les z_j sont, en valeurs relatives inférieures à 10^{-8} ; la plupart du temps, l'algorithme converge en moins de 5 itérations.

La routine `rk_imps` est appelée par `evol`, *cf.* §8.55 (Page 167), qui gère l'évolution temporelle de la composition chimique.

7.5.5 Conservation du nombre de nucléons

Pour chaque isotope d'indice i, les équations d'évolution de la composition chimique sont de la forme générale :

$$\frac{dX_i}{dt} = \dot{X}_i = \sum_{j=1}^{m} a^i_j X_j + \sum_{\substack{j=1\\k\geq j}}^{m} a^i_{jk} X_j X_k + \sum_{\substack{j=1\\l\geq k\geq j}}^{m} a^i_{jkl} X_j X_k X_l$$

Les abondances X_i des m différentes espèces chimiques satisfont des relations linéaires de conservation (nombre de nucléons, charges) de la forme :

$$\sum_{i=1}^{m} \gamma_i X_i = cte \Rightarrow \frac{\partial}{\partial t}\sum_{i=1}^{m} \gamma_i X_i = 0 = \sum_{i=1}^{m} \gamma_i \dot{X}_i.$$

(Cette dernière relation constitue d'ailleurs un test analytique qui permet de vérifier qu'il n'y a pas d'erreur dans le formalisme des équations d'évolution.)

Un algorithme linéaire d'intégration permet de conserver parfaitement de telles relations. Il est de la forme générale *cf.* Henrici [1962] :

$$y_n = \sum_{i=1}^{k} \alpha_i y_{n-i} + h\sum_{i=0}^{k} \beta_i y'_{n-i}, \quad \sum_{i=1}^{k} \alpha_i = 1,\ n \geq k$$

où α_i et β_i, $i = 1, \ldots, n$ sont des constantes, h le pas d'intégration, y_n le vecteur des abondances des espèces chimiques à l'intant $t_0 + hn$, t_0 l'instant initial et $k + 1$ le nombre de points sur lesquels s'appuie l'algorithme $1 \leq k \sim 7$.

Avec un algorithme linéaire, il y a conservation parfaite d'une combinaison linéaire des X_j puisque :

$$\begin{aligned}
&\textstyle\sum_{j=1}^{m} \gamma_j X_{j,n} = \sum_{j=1}^{m} \gamma_j \sum_{i=1}^{k} \alpha_i X_{j,n-i} + \sum_{j=1}^{m} \gamma_j h \sum_{i=0}^{k} \beta_i \dot{X}_{j,n-i} =\\
&\textstyle= \sum_{i=1}^{k} \alpha_i \sum_{j=1}^{m} \gamma_j X_{j,n-i} + \sum_{i=0}^{k} \beta_i h \sum_{j=1}^{m} \gamma_j \dot{X}_{j,n-i} =\\
&= cte + 0.
\end{aligned}$$

7.5.6 Conservation des baryons et de la charge

A l'instant t, pour un nombre d'isotopes $n_{\rm elem} > 1$, les expressions de la conservation du nombre de baryons et de la charge, ont la forme générale d'une relation linéaire entre les abondances, avec des coefficients α_i ***constants*** :

$$\sum_{i=1}^{n_{\rm elem}} \alpha_i x^t_i = cte,$$

alors :

$$\frac{\partial}{\partial t}\sum_{i=1}^{n_{\rm elem}} \alpha_i x_i = 0 \Longrightarrow \sum_{i=1}^{n_{\rm elem}} \alpha_i \Psi_i = 0.$$

en tenant compte de la linéarité de Eq. 7.18 (Page 88) et Eq. 7.23 (Page 89), avec des notations évidentes :

$$\sum_{i=1}^{n_{\rm elem}} \alpha_i z_i = 0,$$

alors :

$$\sum_{i=1}^{n_{\rm elem}} \alpha_i x_i^{t+dt} = \sum_{i=1}^{n_{\rm elem}} \alpha_i x_i^t + \sum_{i=1}^{n_{\rm elem}} \alpha_i z_i = cte;$$

en particulier, avec le schémas IRK LobattoIIIC, le nombre de baryons et de charges est conservé au niveau :

- de la fermeture du schéma de Newton-Raphson,
- de l'ordre du schéma d'intégration utilisé pour le mélange convectif,
- des erreurs d'arrondi.

Des tests numériques ont montré, dans le cas solaire, que sur un pas temporel, le nombre de baryons était conservé à mieux que 10^{-13} en valeur relative *cf.* Morel [1997] .

7.5.7 Normalisation de la somme des abondances

Bien que les algorithmes utilisés pour la résolution numérique des équations d'évolution temporelle des espèces chimiques assurent la plupart des relations de conservation, ils ne peuvent le faire qu'à la précision requise. Pour limiter les conséquences de ces approximations, une normalisation de la somme des abondances par unité de masse est effectuée de façon à assurer la relation $X + Y + Z = 1$. La normalisation est effectuée dans la routine `evol`, *cf.* § 8.55 (Page 167) à l'issue de l'intégration temporelle de la composition chimique.

L'abondance par masse de l'élément chimique s'écrit $X_i = x_i a_i$ où x_i est l'abondance par mole et a_i est la masse atomique de l'élément. On a donc : $X + Y + Z = \sum_i x_i \nu_i$. Avec le développement sur la base de B-splines d'ordre m, au point d'abscisse ν, $\nu_{l-1} \le \nu < \nu_l$, l'abondance par mole s'écrit :

$$x_i = \sum_{j=l-m+1}^{l} x_{ij} N_j^m(\nu).$$

et de là :

$$\sum_i X_i = \sum_i a_i \sum_{j=l-m+1}^{l} x_{ij} N_j^m(\nu) = \sum_{j=l-m+1}^{l} N_j^m(\nu) \sum_i a_i x_{ij}. \tag{7.25}$$

On obtient une normalisation des abondances par unité de masse X_i en posant :

$$X_i = x'_i a_i, \quad x'_i = \sum_{j=l-m+1}^{l} x'_{ij} N_j^m(\nu), \quad x'_{ij} = \frac{x_{ij}}{\eta_j}, \quad \eta_j = \sum_i a_i x_{ij}.$$

La somme des B-splines en tout point ν étant égale à l'unité, on obtient la normalisation désirée :

$$X + Y + Z = \sum_{j=l-m+1}^{l} N_j^m(\nu) \sum_i a_i x'_{ij} = \sum_{j=l-m+1}^{l} N_j^m(\nu) \sum_i a_i \frac{x_{ij}}{\eta_j} =$$

$$\sum_{j=l-m+1}^{l} N_j^m(\nu) \frac{1}{\eta_j} \sum_i a_i x_{ij} = \sum_{j=l-m+1}^{l} N_j^m(\nu) = 1.$$

7.5.8 Estimation de la précision de l'intégration

Malgré un grand nombre d'essais, il n'a pas été possible d'utiliser valablement une méthode numérique d'estimation de la précision de l'intégration, la raison étant principalement un coût excessif en calcul de ces méthodes ; dans CESAM2k la précision est assurée par la limitation de la variation temporelle relative des abondances des éléments que l'utilisateur désire contrôler.

7.6 Evolution du moment cinétique sans diffusion

CESAM2k ne considère qu'une dimension radiale ; pour les modèles en rotation, on tient compte de la valeur moyenne de l'accélération centrifuge sur la calotte sphérique de rayon r.

Compte tenu de l'hypothèse de sphéricité, le moment d'inertie $d\mathcal{I}$ du volume sphérique élémentaire $dV = r\sin\theta dr d\theta d\phi$ au point de coordonnées sphériques (r,θ,ϕ) est :

$$d\mathcal{I} = r\sin\theta \rho r dr d\theta d\phi.$$

Le moment d'inertie par unité de masse de la calotte sphérique s'écrit :

$$\frac{dI}{dm} = \frac{\int_0^{2\pi} d\mathcal{I}d\phi}{dm} = \frac{2\pi r^4\rho dr}{4\pi r^2\rho dr}\int_0^\pi \sin^2\theta d\theta = \frac{2}{3}r^2.$$

Le moment cinétique spécifique *i.e.* le moment cinétique par unité de masse est alors :

$$\mathcal{M}_\Omega(t) = \frac{2}{3}r^2\Omega(t).$$

$\Omega(t)$ est la vitesse angulaire moyenne de la calotte sphérique au temps t.

L'hypothèse de ***conservation locale du moment cinétique spécifique*** s'écrit :

$$\frac{d\mathcal{M}_\Omega}{dt} = \frac{dr^2\Omega}{dt} = 0.$$

Avec l'hypothèse de rotation rigide, le moment cinétique total change au cours de l'évolution, en raison des variations locales de densité et de rayon. On peut imaginer qu'un processus physique, non décrit, permet de conserver globalement le moment cinétique, auquel cas la vitesse angulaire *rigide* varie au fur et à mesure de l'évolution[14]. Compte tenu de l'hypothèse de sphéricité, le moment cinétique total au temps t est :

$$\mathcal{M}_\Omega(t) = \frac{2}{3}\int_0^{M_\star(t)} r(t,m)^2\Omega(t,m)\,\mathrm{d}m,$$

où $\Omega(t,m)$ est la vitesse angulaire locale au temps t pour l'abscisse lagrangienne m. Avec l'hypothèse de rotation rigide i.e. $\Omega(t,m) \equiv \Omega(t)$ et conservation globale du moment cinétique i.e. $\mathcal{M}_\Omega(t) \equiv \mathcal{M}_\Omega = cte.$, la vitesse angulaire $\Omega(t)$ à l'instant t vérifie :

$$\mathcal{M}_\Omega = \frac{2}{3}\Omega(t)\int_0^{M_\star(t)} r(t,m)^2\,\mathrm{d}m.$$

[14]La possibilité de la conservation globale du moment cinétique a été implantée dans CESAM2k à l'initiative de MJ.Goupil.

L'énergie cinétique spécifique de rotation est :

$$\epsilon_\Omega = \frac{1}{2}\frac{dI}{dm}\Omega^2 = \frac{1}{3}r^2\Omega^2.$$

Au cours de l'évolution, la vitesse angulaire locale varie en raison de la perte de moment cinétique, ou de la conservation locale ou globale du moment cinétique ; ces variations conduisent à celles de la vitesse aérolaire qui ont pour conséquences des variations de l'énergie cinétique spécifique locale ϵ_Ω :

$$\frac{\partial \epsilon_\Omega}{\partial t} = \frac{1}{3}\frac{\partial (r\Omega)^2}{\partial t} \simeq \frac{1}{3}\frac{r^2(t+\Delta t)\Omega^2(t+\Delta t) - r(t)^2\Omega(t)^2}{\Delta t}$$

On suppose que ces variations d'énergie sont, selon leur signe, prises ou restituées au milieu sous la forme de sources de chaleur. L'équation d'énergie, Eq. 7.3 (Page 66), estimée dans la routine `static_m` ou `static_r` *cf.* § 8.123 (Page 204), devient alors :

$$\frac{\partial L}{\partial M} = \epsilon - \frac{\partial U}{\partial t} + \frac{P}{\rho^2}\frac{\partial \rho}{\partial t} - \frac{\partial \epsilon_\Omega}{\partial t}.$$

Avec une rotation solide de vitesse angulaire non nulle, le moment cinétique est réestimé pour chaque modèle intermédiaire dans la routine `resout`, *cf.* § 8.114 (Page 195).

7.7 Evolution temporelle avec diffusion

7.7.1 Intégration par éléments finis

En raison du mélange convectif, les équations de la diffusion des éléments chimiques et du moment cinétique forment un problème intégro-différentiel aux limites du quatrième ordre avec conditions initiales. Bien que ces équations soient couplées, CESAM2*k* les résout séparément en alternance (méthode des lignes) :

1. La diffusion du moment cinétique, problème du quatrième ordre.
2. La diffusion des éléments chimiques, problème différentiel parabolique.

CESAM2*k* utilise la méthode des éléments finis pour leur intégration numérique. Formellement ces équations sont de la forme :

$$a\frac{Dy}{Dt} + bG = \frac{\partial F}{\partial x} \tag{7.26}$$

$F(x,y)$, $a(x,y)$ et $b(x,y)$ étant des fonctions de l'abscisse x et de la fonction inconnue $y(x)$.

Comme il a été évoqué précédemment, les fonctions inconnues sont projetées sur une base fonctionnelle de dimension finie. Les projections sont obtenues en écrivant que les développements satisfont les équations différentielles et les conditions limites. Pour ces fonctions de base, CESAM2*k* utilise des polynômes par morceaux d'ordre m (polynômes de degré $m-1$), se raccordant avec une oscularité de classe $\mathcal{C}_i$, $0 \le i \le m-1$, (continuité des dérivées d'ordre 0 à $m-1$). Ces polynômes par morceaux sont exprimés sous la forme de B-splines polynômiales normalisées d'ordre

m et de dimension d : $\{S_j^m\}_{j=1}^d$. Ainsi chacune des fonctions est approchée par une fonction d'interpolation de la forme :

$$f(x) \simeq \sum_{j=1}^{d} f_j S_j^m(x) \text{ donc } : \frac{df}{dx} \simeq \sum_{j=1}^{d} f_j \frac{dS_j^m}{dx}.$$

la connaissance des f_j détermine la fonction d'interpolation de f avec une précision que l'on sait estimer *cf.* Schumaker [1981]. Des algorithmes stables et précis permettent de calculer les $S_j^m(x)$ ainsi que toutes leurs dérivées, objet de la routine `bvald`, *cf.* §9.2.6 (Page 216), de CESAM2k.
Le principe de la résolution des équations de diffusion consiste à imposer à la base d'être orthogonale à son image par l'opérateur différentiel, *i.e.* orthogonale au résidu. C'est la "méthode de Galerkin". Son application aux cas de la diffusion des espèces chimiques et du moment cinétique est "semi-spectrale", les fonctions de base ne vérifiant pas les conditions limites. Les solutions numériques obtenues sont dites *faibles*. En formant les produits scalaires :

$$< f \bullet g >= \int_{x_0}^{x_1} f g \, dx \tag{7.27}$$

on obtient :

$$< (a\frac{Dy}{Dt} + bG) \bullet S_j^m >=< \frac{\partial F}{\partial x} \bullet S_j^m >$$

En intégrant par parties le second produit scalaire on obtient :

$$< \frac{\partial F}{\partial x} \bullet S_j^m >= [F S_j^m]_{x_0}^{x_1} - < F \bullet \frac{\partial S_j^m}{\partial x} >$$

On a :

$$[F]_{x_0}^{x_1} \equiv [F S_j^m]_{x_0}^{x_1} = \begin{cases} -F(x_0) & \text{si } j = 1 \\ F(x_1) & \text{si } j = d \\ 0 & \text{sinon} \end{cases}$$

puisqu'aux extrémités de l'intervalle d'intégration, une seule des B-splines est non nulle et égale à l'unité. *In fine* on obtient la discrétisation suivante :

$$< \left(a\frac{y^{t+1} - y^t}{\Delta t} + bG\right) \bullet S_j^m > + < F \bullet \frac{\partial S_j^m}{\partial x} > -[F]_{x_0}^{x_1} = 0 \tag{7.28}$$

On est ramené - éventuellement après linéarisation - à la résolution d'un système linéaire ; les B-splines étant à support borné, le système linéaire est bande-diagonal. Il est résolu par élimination de Gauss (pivots partiels) par la routine `gauss_dp`, *cf.* §9.1 (Page 213). Les produits scalaires sont approchés par une intégration numérique de type Gauss, suivant un algorithme inspiré de l'algorithme 5.22, p.203 de Schumaker [1981].

La nature "intégrale" des équations résulte du mélange convectif. Il est effectué dans CESAM2k par le biais d'une diffusion turbulente de coefficient $D_{\rm M} >> 1$.

Ces dispositions ont les avantages suivants :
– L'ordre des équations différentielles à résoudre est réduit d'une unité.

– Les discontinuités des dérivées premières sont implicitement prises en compte par la formulation intégrale, Eq. 7.38 (Page 97), des produits scalaires.
– Avec l'utilisation de la base des B-splines les algorithmes restent simples même lorsqu'il se produit des déplacements des limites zone radiative / zone convective.

7.7.2 Bases de B-splines pour Petrov-Galerkin

Une représentation schématique du vecteur nodal pour[15] $m = 4$, avec discontinuité de la ***dérivée première*** au point fléché $\Downarrow$, *cf.* Schumaker [1981], est la suivante :

$$\begin{array}{cccccccccc}
 & & & & & \Downarrow & & & & \\
\times & \times & \times & \dots & \times & \times & \times & \dots & \times & \times \\
\times & & & & & \times & & & & \times \\
\times & & & & & \times & & & & \times \\
\times & & & & & & & & & \times
\end{array}$$

Avec continuité de la fonction et des deux premières dérivées, en tous les points de grille :

$$\begin{array}{cccccccccc}
\times & \times & \times & \dots & \times & \times & \times & \dots & \times & \times \\
\times & & & & & & & & & \times \\
\times & & & & & & & & & \times \\
\times & & & & & & & & & \times
\end{array}$$

7.7.3 Présence de discontinuités

Dans le cas où la fonction à intégrer présente des discontinuités, la base de B-splines sur laquelle elle est projetée, doit inclure ces discontinuités. Avec une discontinuité au point x_d de $]x_0, x_1[$ l'intégrale du produit scalaire, *cf.* équation 7.27 (Page 94), est définie par :

$$< f \bullet g >= \int_{x_0}^{x_{d^-}} f g \, dx + \int_{x_{d^-}}^{x_1} f g \, dx, \tag{7.29}$$

Pour le problème différentiel, *cf.* équation 7.26 (Page 93), avec le produit scalaire ainsi défini, l'intégration par parties est justifiée, car elle ne porte que sur des fonctions continues. *In fine* pour les B-splines dont le support est contenu dans l'intervalle $[x_0, x_{d^-}[$ on obtient :

$$< \left(a\frac{y^{t+1} - y^t}{\Delta t} + bG \right) \bullet S_j^m > + < F \bullet \frac{\partial S_j^m}{\partial x} > -[F]_{x_0}^{x_{d^-}} = 0, \tag{7.30}$$

et pour celles dont le support est contenu dans l'intervalle $[x_1, x_{d^+}[$:

$$< \left(a\frac{y^{t+1} - y^t}{\Delta t} + bG \right) \bullet S_j^m > + < F \bullet \frac{\partial S_j^m}{\partial x} > -[F]_{x_{d^+}}^{x_1} = 0. \tag{7.31}$$

Les produits scalaires étant définis par l'équation 7.29 (Page 95).
En un point de discontinuité le vecteur nodal présentera m abscisses identiques.

[15] Polynômes par morceaux de degré 3, se raccordant par la continuité de leurs dérivées secondes.

Pour un ordre de B-spline $m = 4$, une représentation schématique du vecteur nodal avec discontinuité au point fléché est :

$$\begin{array}{cccccccccc}
 & & & & & \Downarrow & & & & \\
\times & \times & \times & \ldots & \times & \times & \times & \ldots & \times & \times \\
\times & & & & & \times & & & & \times \\
\times & & & & & \times & & & & \times \\
\times & & & & & \times & & & & \times
\end{array}$$

Cette disposition est équivalente à autant de domaines d'intégration distincts qu'il y a de discontinuités. Cette représentation "regroupée" n'est utile que si les discontinuités apparaissent, disparaissente ou encore se déplacent. Certains systèmes différentiels peuvent présenter simultanément, en divers points du domaine d'intégration des variables continues, des variables continues non dérivables et des variables discontinues. Un exemple est celui du système des équations de diffusion du moment cinétique, *cf.* § 7.9 (Page 111). Une représentation schématique d'un vecteur nodal avec une discontinuité de la dérivée première ($\downarrow$) et une discontinuité ($\Downarrow$) est :

$$\begin{array}{ccccccccccc}
 & & \downarrow & & & & \Downarrow & & & & \\
\times & \times & \times & \times & \ldots & \otimes & \times & \times & \ldots & \times & \times \\
\times & & \times & & & & \times & & & & \times \\
\times & & \times & & & & \times & & & & \times \\
\times & & & & & & \otimes & & & & \times
\end{array}$$

Quand le problème différentiel présente, en des abscisses identiques, des variables continues et des variables discontinues[16], CESAM$2k$ assure la continuité en égalant les projections des variables concernées sur chacune des deux B-splines localisées de part et d'autre la discontinuité. Dans la représentation schématique précédente, ces B-splines sont identifiées avec le symbole $\otimes$. Pour obtenir autant d'équations que d'inconnues la relation résultant du produit scalaire avec une des B-splines concerné est supprimé. Suppression qui élimine la contribution de cette B-spline à la quantité intégrée, *cf.* § 7.7.1 (Page 93).

7.8 Diffusion des éléments chimiques

L'équation d'évolution temporelle des espèces chimiques avec diffusion :

$$\frac{Dx_i}{Dt} = \frac{\partial F_i}{\partial m} + \Upsilon_i,\ 1 \le i \le n_{\rm elem} \tag{7.32}$$

contient une partie diffusive, $\frac{\partial F_i}{\partial m}$, et une partie nucléaire, Υ_i. Cette dernière est calculée par la routine de type `nuc` utilisée, *cf.* § 8.76 (Page 179). Le flux de particules d'abondance en nombre x_i a pour expression :

$$F_i = 4\pi R^2\rho \left(4\pi R^2 \rho D_i * \nabla_{\rm m}\mathcal{X} - v_i x_i\right), \tag{7.33}$$

les composantes $d_{i,j},\ i \ne j$, du vecteur :

$$D_i(m,t) \equiv (d_{i,1}, \ldots, d_{i,n_{\rm elem}})^T,$$

[16]Tel sera le cas pour le système des équations de diffusion du moment cinétique.

sont les coefficients, $d_{i,j} = d^\star_{i,j}$, de diffusion microscopique de l'élément d'indice i, par rapport à l'élément d'indice j ; la (i,i)-ième composante :

$$d_{i,i} \equiv d^\star_{i,i} + d_\mathrm{T} + d_\mathrm{M} \tag{7.34}$$

inclut les coefficients de diffusion turbulente d_T et de mélange d_M ; $v_i(m,t)$ est la vitesse de sédimentation ; on a noté à l'aide du symbole "$*$" le produit scalaire :

$$D_i * \nabla_\mathrm{m}\mathcal{X} \equiv \sum_j d_{i,j}\frac{\partial x_j}{\partial m}.$$

En utilisant la variable spatiale $\nu \equiv (m/M_\odot)^{2/3}$, l'équation 7.32 (Page 96) devient :

$$\frac{\partial x_i}{\partial t} = -\frac{\partial F_i}{\partial m} + \Upsilon_i = -\frac{2}{3M_\odot\sqrt{\nu}}\frac{\partial F_i}{\partial \nu} + \Psi_i,\ 1 \le i \le n_\mathrm{elem}. \tag{7.35}$$

Des zones mélangées, *i.e.* zones convectives éventuellement overshootées, apparaissent, se déplacent, disparaissent au cours de l'évolution. On admet que le mélange convectif est suffisamment efficace pour uniformiser la composition chimique, ainsi :

$$\frac{\partial x_i}{\partial R} = 0. \tag{7.36}$$

Ces conditions entraînent la discontinuité des gradients des abondances aux limites entre zones radiatives et zones mélangées ; d'où la nécessité d'introduire des limites internes mobiles et en nombre variable, ce qui complique l'intégration numérique. Grâce à un choix *ad hoc* des bases de B-splines utilisées pour la diffusion des éléments chimiques, les discontinuités des gradients sont implicitement prises en compte.

La présence du mélange convectif change la nature du problème, de différentiel il devient ***intégro-différentiel***. L'équation de transport des éléments chimiques étant du type advection-diffusion, on contourne l'aspect intégral en uniformisant la composition chimique par une diffusion. Pour ce faire, on impose dans les zones mélangées un coefficient de diffusion dont l'ordre de grandeur, grand devant l'unité, correspond au temps de retournement des éléments convectifs. CESAM2k utilise $d_\mathrm{conv} = 10^{13}$ cm^{2}s^{-1}. A chaque limite entre une zone mélangée et la zone non mélangée adjacente les coefficients de diffusion sont discontinus avec pour conséquence la discontinuité des gradients des abondances des espèces chimiques.

On représente l'abondance de chaque espèce par un polynôme par morceaux d'ordre $m_c \ge 2$ ***avec discontinuité de la dérivée première*** à chaque limite zone radiative / zone convective, *cf.* §7.2.1 (Page 67). Pour le calcul numérique, on projette ces polynômes sur une base de B-splines avec discontinuité de la dérivée première : $\mathcal{N} \equiv \{N^{m_c}_k\}_{k=1,K_\mathrm{X}}$, $\mathcal{N} \subset \mathcal{C}^0[0,\nu_b]$; K_X étant la dimension de la base. On a alors :

$$x_i(\nu,t) = \sum_{k=1}^{K_\mathrm{X}} x_{i,k}(t)N^{m_c}_j(\nu). \tag{7.37}$$

On recherche les *solutions faibles* de l'équation 7.35 (Page 97) à l'aide du formalisme de Galerkin. Les produits scalaires s'écrivent :

$$< \frac{3}{2}M_\odot\sqrt{\nu}\left(\frac{\partial x_i}{\partial t} - \Psi_i\right) \bullet S^{m_c}_j > + < \frac{\partial F_i}{\partial \nu} \bullet S^{m_c}_j >= 0, \tag{7.38}$$

En tenant compte de la continuité des flux F_i, l'intégration par parties conduit à :

$$< \frac{\partial F_i}{\partial \nu} \bullet S_j^{m_c} >= \left[F_i S_j^{m_c}\right]_0^{\nu_\mathrm{b}} - < F_i \bullet \frac{\partial S_j^{m_c}}{\partial \nu} >, \tag{7.39}$$

avec la condition limite, $\nu \equiv 0$, $F_i(\nu = 0) = 0$ au centre et, à la limite externe de l'enveloppe $\nu = \nu_\mathrm{b}$. On montre au § 7.8.1 (Page 98) :

$$F_i(\nu = \nu_\mathrm{b}) = \dot{M}(x_{iv} - x_i)$$

où x_{iv} est l'abondance par mole de l'espèce i dans le vent stellaire. Par convention, le taux de perte de masse $\dot{M}$ est négatif s'il y a perte, et positif s'il y a gain. Comme à la limite de l'enveloppe, seule la dernière spline $S_{K_X}^{m_c}$, est non nulle et comme $S_{K_X}^{m_c}(\nu_\mathrm{b}) \equiv 1$ la quantité intégrée devient :

$$\left[F_i S_j^{m_c}\right]_0^{\nu_\mathrm{b}} = \dot{M}(x_{iv} - x_i)(\nu_\mathrm{b}).$$

Physiquement, le vent stellaire ne peut concerner que la limite externe, à défaut de le décrire en détail, on suppose qu'il concerne toute la zone convective externe de masse M_ZC. Pour simplifier l'algorithme on transpose la partie intégrée dans le terme nucléaire Ψ_i ainsi qu'il est décrit au § 7.8.1 (Page 98).
La continuité de x_i et de F_i permet d'écrire l'équation 7.38 (Page 97) sous la forme utilisée dans CESAM$2k$:

$$\begin{aligned} & < \frac{3}{2}\sqrt{\nu}\left(x_i - x_i^{(t)} - \Psi_i \Delta t\right) \bullet S_j^{m_c} > - \frac{\dot{M}}{M_\mathrm{ZC}}(x_{iv} - x_i)\Delta t + \\ + & < \left[\frac{32\pi^2 R^4 \rho^2 \,\Delta t}{3M_\odot \sqrt{\nu}}\left\{D_i * \nabla_\nu \mathcal{X}\right\} - \frac{4\pi R^2 \rho}{M_\odot}\Delta t\, V_i x_i\right] \bullet \frac{\partial S_j^{m_c}}{\partial \nu} >= 0,\ j = 1, \ldots, K_X, \end{aligned} \tag{7.40}$$

les quantités étant prises à l'instant $t + \Delta t$ sauf $x_i^{(t)}$ pris au temps t. Ce schéma complètement implicite est d'ordre m_c en espace, et du premier ordre en temps.
Au centre, par symétrie, les dérivées spatiales des abondances sont nulles ainsi que leurs vitesses d'advection. Il en est de même pour la limite externe toujours située dans une zone de mélange. Le terme intégré qui est nul a été omis dans l'écriture des équations.

7.8.1 Condition limite externe

On désignera par "vent", indistinctement un apport ou un retrait de masse. A défaut de décrire le processus de création du vent, on suppose que celui-ci est issu de la zone convective externe qui existe toujours. Pendant l'intervalle de temps Δt, l'apport en masse d'élément chimique i à la zone convective $\dot{M} X_{iv} \Delta t$, X_{iv} étant l'abondance par masse de l'élément i au temps t dans le vent.

A l'issue du pas temporel, la fraction de masse de l'élément i dans la zone convective sera :

$$X_i^{t+\Delta t} = \frac{M_\mathrm{ZC} X_i^t + \dot{M} X_{iv} \Delta t}{\sum_i M_\mathrm{ZC} X_i^t + \dot{M} X_{iv} \Delta t},$$

où X_i^t et X_{iv} sont respectivement l'abondance par masse de l'élément i au temps t et dans le vent. Les fractions de masse vérifiant $\sum_i X_i^t \equiv 1$ et $\sum_i X_{iv}^t \equiv 1$:

$$X_i^{t+\Delta t} = \frac{M_{\rm ZC} X_i^t + \dot M X_{iv}\Delta t}{M_{\rm ZC} + \dot M \Delta t} = \frac{X_i^t + \frac{\dot M}{M_{\rm ZC}} X_{iv}\Delta t}{1 + \frac{\dot M}{M_{\rm ZC}}\Delta t} \sim$$
$$X_i^t\left(1 - \frac{\dot M}{M_{\rm ZC}}\Delta t\right) + \frac{\dot M}{M_{\rm ZC}} X_{iv}\Delta t \left(1 - \frac{\dot M}{M_{\rm ZC}}\Delta t\right) =$$
$$X_i^t - \frac{\dot M}{M_{\rm ZC}} X_i^t \Delta t + \frac{\dot M}{M_{\rm ZC}} X_{iv}\Delta t - \left(\frac{\dot M}{M_{\rm ZC}} X_{iv}\Delta t\right)^2 \sim X_i^t + \frac{\dot M}{M_{\rm ZC}}\left(X_{iv} - X_i^t\right)\Delta t$$

On obtient pour la dérivée temporelle de l'abondance de l'espèce chimique i :

$$\dot X_i = \lim_{\Delta t \to 0} \frac{X_i^{t+\Delta t} - X_i^t}{\Delta t} = \frac{\dot M}{M_{\rm ZC}}\left(X_{iv} - X_i^t\right) \iff \dot x_i = \frac{\dot M}{M_{\rm ZC}}\left(x_{iv} - x_i^t\right), \qquad (7.41)$$

où x_{iv} est l'abondance par mole de l'élément i dans le vent. Dans la zone convective externe la variation temporelle de l'isotope X_i est donné par :

$$\dot x_i M_{\rm ZC} = \dot M (x_{iv} - x_i^t). \qquad (7.42)$$

Les abondances ne varient que si la composition chimique du vent est différente de celle des couches externes. Dans la zone convective externe les variations de composition chimique résultant de l'équation 7.42 sont appliquées en addition de celles résultant des réactions thermonucléaires.
Ces dispositions sont appliquées même si la diffusion microscopique des éléments chimique est ignorée.

7.8.2 Chutes de planètoïdes

De façon similaire à ce qui précède, au cours d'une évolution CESAM2k permet de simuler une chute de planètoïdes sur un intervalle de temps limité. La composition chimique de la zone convective externe se trouve alors modifiée par des éléments chimiques dont l'abondance et la nature peuvent, éventuellement, différer de celles des conditions initiales et/ou locales. Dans la zone convective externe la variation temporelle de l'isotope X_i est donné par :

$$\dot x_i(t) = \frac{N_{\rm P} M_\oplus}{M_{\rm ZC}} \mathcal{P}(t)(x_{iv} - x_i^t). \qquad (7.43)$$

$N_{\rm P} > 0$ est le nombre total de planètoïdes de masse terrestre $M_\oplus$ reçus par l'étoile, $\mathcal{P}(t)$ la fonction décrivant la dépendance temporelle, *cf.* § 4.6 (Page 41).

CESAM2k peut tenir compte de l'apport de moment cinétique à la zone convectiveexterne résultant des chutes de planètoïdes, *cf.* § 7.9.18 (Page 125).

7.8.3 Formalisme de Burgers

Le formalisme de Burgers [1969], permet de déterminer les coefficients de diffusion microscopique des éléments chimiques. On reformule le système des équations de Burgers pour les variables utilisées par CESAM2k, les abondances par mole x_i et le poids moléculaire moyen μ. Le principe de la méthode numérique de Burgers, Iben & MacDonald [1985], consiste à remarquer que la résolution d'un système linéaire permet d'écrire la vitesse de diffusion w_i de la particule i sous la forme :

$$w_i = v_i + \sum_j b_{ij} \frac{dx_j}{dx}.$$

Iben & MacDonald [1985] n'ont pas tenu compte des "residual heat flow vector" qui furent introduits par la suite par Cox *et al.* [1989] et repris par Thoul *et al.* [1994].

On utilise les notations suivantes :

- c : nombre d'ions, il y a $c+1$ espèces de particules, *i.e.* ions+ électrons,
- E : champ électrique,
- e : charge de l'électron, en unité électrostatique, et indice de la particule "électron",
- $g_{\rm G}$: accélération due à la gravité,
- g_Ω : accélération centrifuge,
- $g_{\rm e}$: accélération effective,
- $g_{\rm Ri}$: accélération radiative sur la particule i,
- g_i : accélération due à la gravité effective et aux forces radiatives sur i,
- $g_{\rm t}$: accélération totale,
- $g_{j,i}$: poids statistique du fondamental du niveau d'ionisation j de l'élément chimique i,
- $m_e = \nu_e m_u$: masse de l'électron,
- $m_i \equiv \nu_i m_u$: masse de l'élément chimique i,
- $m_{ij} \equiv \frac{m_i m_j}{m_i + m_j}$: masse réduite des particules d'indices i et j,
- $m_u = 1/N_0$: masse atomique unité, inverse du nombre d'Avogadro,
- n_e : nombre d'électrons par unité de volume,
- n_i : nombre d'ions de l'élément chimique $i \neq e$ par unité de volume,
- $n_{j,i}$: nombre d'ions de l'élément chimique i par unité de volume, dans l'état d'ionisation $j = 0, \ldots, Z_i$,
- r : rayon, variable d'espace eulérienne,
- $T,\ P,\ \rho$: température, pression gaz parfait, densité,
- t : temps,
- $\mathcal{V} = \frac{\partial r}{\partial t}$: vitesse d'entraînement du fluide, $\mathcal{V} \sim 0$ (hypothèse de l'équilibre quasi-statique),
- w_i : vitesse de diffusion de la particule i, par rapport à la vitesse moyenne du fluide,
- X_i : proportion en masse de l'abondance de l'élément chimique i, $\sum_i X_i \equiv 1$,
- $x_i \equiv X_i/\nu_i$: proportion par mole de l'abondance de l'élément chimique i,
- $x_{j,i}$: taux d'ionisation du niveau d'ionisation j de l'ion i,
- Z_i charge de la particule i, $Z_e = -1$,
- $\bar{Z}_i$: charge moyenne de l'élément i avec ionisation,
- η : paramètre de dégénérescence,
- μ : poids moléculaire moyen,
- ν_e : "masse atomique" de l'électron, $\nu_e = m_e/m_u$,
- ν_i : masse atomique de l'élément chimique i,
- $\chi_{j,i}$: potentiel d'ionisation du niveau d'ionisation j de l'élément i.

PB `Les notations ne sont pas totalement cohérentes en ce qui concerne les électrons.`

7.8.4 Charge moyenne des ions

On utilise l'équation de Saha (Cox & Guili [1968], formule 15-30) dans laquelle on limite les fonctions de partition aux poids statistiques des niveaux fondamentaux :

$$\frac{n_{j-1,i}}{n_{j,i}} = \frac{g_{j-1,i}}{g_{j,i}} \exp\left(\eta - \Delta\mu + \frac{\chi_{j,i}}{kT}\right) \equiv \varphi_{j,i},\ j = 1, \ldots, Z_i. \tag{7.44}$$

Le taux de dégénérescence η vérifie :

$$F_{\frac{1}{2}}(\eta) = \frac{n_e}{4\pi}\left(\frac{h^2}{2m_e kT}\right)^{\frac{3}{2}},$$

$F_{\frac{1}{2}}(\eta)$ est la fonction de Fermi-Dirac (Clayton [1968], formule 2-57). $\Delta\mu$ est une correction numérique introduite par Eggleton *et al.* [1973], pour simuler l'ionisation de pression, et éviter la recombinaison à haute température (Kippenhahn & Weigert [1991] §14.6) :

$$\Delta\mu = n_e c_F \left(\frac{a_0}{\bar Z}\right)^3 (1 + \frac{20\chi_0}{kT}),\ a_0 = \frac{h^2}{4\pi^2 m_e e^2},\ \bar Z = X + 2Y + 8Z,\ c_F = 15,\ \chi_0 = 13.6.$$

PB μ `n'est pas, dans ce paragraphe, le poids moléculaire moyen.`

L'expérience des calculs a montré que le formalisme d'Eggleton *et al.* [1973], même avec la modification de Proffit & Michaud [1991] formule (4), donnait des taux d'ionisation moyens avec des paliers et des gradients beaucoup plus marqués que ceux obtenus avec une meilleure prise en compte de l'ionisation de pression. Ainsi, au centre du Soleil, le formalisme d'Eggleton *et al.* [1973] donne un taux d'ionisation de 100% pour Fe XXV, alors que le calcul plus précis de Gabriel [1997] ne donne que 85%.

Pour tenir compte de l'ionisation de pression et éviter la recombinaison, la routine `saha` de CESAM2k utilise une correction numérique dérivée de l'analyse de Clayton [1968] p. 140-145. Le potentiel d'ionisation est réduit lorsque la distance moyenne entre les ions devient de l'ordre de grandeur de la longueur de Debye :

$$R_{\rm D} = \sqrt{\frac{kT}{4\pi e^2 \rho N_0 \zeta}}, \quad \zeta = \sum_{i \neq e} \bar Z_i (\bar Z_i + 1) x_i$$

et se réduit[17] à :

$$\chi'_{j,i} = \max(0, \chi_{i,j} - \chi^{\rm C}_{i,j}), \quad \chi^{\rm C}_{i,j} = \frac{je^2}{R_{\rm D}}.$$

Cette correction appliquée seulement sur le potentiel d'ionisation s'est avérée insuffisante pour éviter la recombinaison à haute température, en particulier, de l'ion HII. La routine `saha` utilise un ajustement *numérique* agissant à la fois sur le potentiel

[17] je est la charge de l'ion.

d'ionisation et le quotient des fonctions de partition ; celles-ci devant être tronquées de façon cohérente avec la diminution des potentiels d'ionisation (*cf.* Clayton [1968] figure 2-16, p. 146). Celà n'est pas possible avec les fonctions de partition limitées aux poids statistiques des niveaux fondamentaux. L'artifice consiste en une diminution progressive de 1 à 0 du rapport des poids statistiques des niveaux fondamentaux $g_{j-1,i}/g_{j,i}$, dès que la quantité :

$$x = \frac{\chi_{i,j}}{\chi'_{j,i}} - 1$$

devient inférieure à une valeur p fixée, pour le niveau d'ionisation $j = 1, \ldots, Z_i$ de l'élément chimique i. L'expérience des calculs a montré que la valeur $p = 4$ permettait de retrouver, pour des modèles d'étoiles de la séquence principale de $1M_\odot$ et $1.4M_\odot$ respectivement, des taux d'ionisation voisins de ceux obtenus avec un calcul plus exact, mais beaucoup plus lourd, de la correction de pression d'ionisation. La fonction de raccordement $f(x)$ utilisée est un morceau de cubique sur $[0, p]$, de pente nulle en $x = 0$ et $x = p$:

$$f(x) = 0 \text{ si } x \leq 0,\ \left(\frac{x}{p}\right)^2 \left[-2\left(\frac{x}{p}\right) + 3\right] \text{ si } x \in [0, p],\ f(x) = 1,\ \text{si } x \geq p.$$

Avec ce formalisme approximatif, l'équation de Saha 7.44 devient :

$$\frac{n_{j-1,i}}{n_{j,i}} = \frac{g_{j-1,i}}{g_{j,i}} f(x) \exp\left(\eta + \frac{\chi_{j,i}}{kT}\right) \equiv \varphi_{j,i},\ j = 1, \ldots, Z_i. \tag{7.45}$$

On obtient ainsi l'ionisation totale du niveau $j-1$ lorsque $g_{j-1,i}/g_{j,i} f(x) = 0$ *i.e.* dès que $\chi_{i,j} \leq \chi^{\mathrm{C}}_{i,j}$.

Le nombre de particules $n_i,\ i \neq e$ est :

$$\begin{aligned}
n_i &= \sum_{j=0}^{Z_i} n_{j,i} = n_{Z_i,i}\left(1 + \sum_{j=0}^{Z_i-1} \frac{n_{j,i}}{n_{Z_i,i}}\right) = n_{Z_i,i}\left(1 + \sum_{j=0}^{Z_i-1} \prod_{k=j+1}^{Z_i} \varphi_{k,i}\right) \\
&= n_{Z_i,i}\left(1 + \frac{n_{0,i}}{n_{Z_i,i}} + \frac{n_{1,i}}{n_{Z_i,i}} + \ \ldots\ + \frac{n_{Z_i-1,i}}{n_{Z_i,i}}\right) \\
&= n_{Z_i,i}\left(1 + \varphi_{1,i}\varphi_{2,i}\varphi_{3,i}\ \ldots\ \varphi_{Z_i,i} + \varphi_{2,i}\varphi_{3,i}\ \ldots\ \varphi_{Z_i,i} +\ \ldots\ + \varphi_{Z_i,i}\right) \\
&= n_{Z_i,i}\left(1 + \varphi_{Z_i,i}\left(1 + \varphi_{Z_i-1,i}\left(1 + \varphi_{Z_i-2,i}\left(1 + \ldots \varphi_{1,i}\right)\right)\right)\right)
\end{aligned}$$

et comme :

$$\varphi_{j+1,i} = \frac{n_{j,i}}{n_{j+1,i}} = \frac{n_{j,i}}{n_i}\frac{n_i}{n_{j+1,i}} = \frac{x_{j,i}}{x_{j+1,i}},$$

on a :

$$\begin{aligned}
x_{Z_i,i} \equiv \frac{n_{Z_i,i}}{n_i} &= \frac{1}{\left(1 + \varphi_{Z_i,i}\left(1 + \varphi_{Z_i-1,i}\left(1 + \varphi_{Z_i-2,i}\left(1 + \ldots \varphi_{1,i}\right)\right)\right)\right)}, \\
x_{j,i} &= \varphi_{j+1,i}\, x_{j+1,i},\ j = Z_i - 1, \ldots, 0.
\end{aligned}$$

La charge moyenne de l'élément chimique i et le nombre d'électrons par mole et par unité de volume s'écrivent respectivement :

$$\bar{Z}_i \equiv \sum_{j=1}^{Z_i} j x_{j,i},\ x_e = \sum_{j \neq e} \bar{Z}_i x_i,\ n_e = \rho N_0 x_e = \rho N_0 \sum_{j \neq e} \bar{Z}_i x_i.$$

On a aussi :

$$X_i = \frac{n_i \nu_i m_u}{\sum_j n_j \nu_j m_u} = \frac{n_i \nu_i}{\rho N_0},\ x_i = \frac{X_i}{\nu_i} = \frac{n_i}{\rho N_0}.$$

7.8.5 Equation de diffusion des espèces chimiques

Pour la particule i, l'équation de diffusion s'écrit (Iben & Mc Donald [1985]) :

$$\frac{\partial n_i}{\partial t} = -\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 n_i w_i) + \left(\frac{\partial n_i}{\partial t}\right)_{\text{nucl.}}$$

et les équations de Burgers [1969] Eq. (18.4) et (18.5), voir aussi Cox *et al.* [1989], Thoul *et al.* [1994] :

$$\frac{dP_i}{dr} - \frac{\rho_i}{\rho}\frac{dP}{dr} - n_i \bar{Z}_i e E = \sum_j K_{ij}(w_j - w_i) + \sum_j K_{ij} z_{ij} \frac{m_j r_i - m_i r_j}{m_i + m_j} \tag{7.46}$$

$$\begin{aligned} \frac{5}{2} n_i k \frac{dT}{dr} = &- \frac{5}{2}\sum_j K_{ij} z_{ij} \frac{m_j (w_j - w_i)}{m_i + m_j} - \frac{2}{5} K_{ii} z''_{ii} r_i \\ &- r_i \sum_{j\neq i} K_{ij} \frac{3m_i^2 + m_j^2 z'_{ij} + \frac{4}{5} m_i m_j z''_{ij}}{(m_i + m_j)^2} \\ &+ \sum_{j\neq i} K_{ij} \frac{m_i m_j (3 + z'_{ij} - \frac{4}{5} z''_{ij})}{(m_i + m_j)^2} r_j. \end{aligned} \tag{7.47}$$

Les équations de conservation dynamique et statique de la masse et de la charge ont respectivement pour expression :

$$\sum_i \nu_i n_i w_i = 0,\ e\sum_i \bar{Z}_i n_i w_i = 0,\ \sum_i \bar{Z}_i n_i = 0,\ \sum_i X_i = \sum_i \nu_i x_i = 1. \tag{7.48}$$

La densité partielle ρ_i de la particule i est définie par :

$$\rho_i = \rho X_i = \rho x_i \nu_i = n_i \nu_i m_u,\ \sum_i \rho_i = \rho = \sum_i n_i \nu_i m_u, \tag{7.49}$$

et :

$$n_i = \rho N_0 \frac{X_i}{\nu_i} = \rho N_0 x_i.$$

Le second terme du membre de gauche de l'équation (7.46) correspond à la force externe qui s'exerce sur la particule i, à savoir, la gravité corrigée de l'accélération radiative.

7.8.6 Intégrales de collision

Les coefficients de résistance du flux de chaleur $z_{ij}, z'_{ij}, z''_{ij}$ sont définis à partir des intégrales de collision $\Omega_{ij}^{(kl)}$, *cf.* Paquette *et al.* [1986] Eq. (23) – (25) :

$$z_{ij} = 1 - \frac{2}{5}\frac{\Omega_{ij}^{(12)}}{\Omega_{ij}^{(11)}},\ z'_{ij} = 2.5 - \frac{2}{5}\frac{5\Omega_{ij}^{(12)} - \Omega_{ij}^{(13)}}{\Omega_{ij}^{(11)}},\ z''_{ij} = \frac{\Omega_{ij}^{(22)}}{\Omega_{ij}^{(11)}},$$

qui sont déterminées à partir de tabulations. Paquette *et al.* [1986] ont écrit $\Omega_{ij}^{(kl)}$ sous la forme, Eq.(65) :

$$\Omega_{ij}^{(kl)} \equiv F_{ij}^{(kl)} \epsilon_{ij},\ \epsilon_{ij} \equiv \frac{e^4}{4}\sqrt{\frac{\pi}{2k^3}}\,\frac{\bar{Z}_i^{\,2}\bar{Z}_j^{\,2}}{\sqrt{m_{ij}}}\,\frac{1}{\sqrt{T^3}}.$$

Pour $(kl) = (11),\ (12),\ (13),\ (22)$, et selon qu'il s'agit d'un potentiel attractif, *i.e.* électron/ion, ou répulsif, *i.e.* ion/ion ou électron/électron, Paquette *et al.* [1986] ont calculé et tabulé les coefficients d'interpolation par splines naturelles de $\ln F_{ij}^{(kl)}$ en fonction de la variable $\psi_{ij} \equiv \ln\left[\ln\left(1+\gamma_{ij}^2\right)\right]$ avec :

$$\gamma_{ij} \equiv \frac{4kT\lambda}{\bar{Z}_i\bar{Z}_j e^2},\ \lambda \equiv \max\left\{\left(\frac{kT}{4\pi e^2 \sum_i n_i \bar{Z}_i}\right)^{\frac{1}{2}}, \left(\frac{3}{4\pi \sum_{\text{ions}} n_i}\right)^{\frac{1}{3}}\right\}.$$

Les quantités K_{ij} sont les coefficients de résistance, ils représentent les effets des collisions entre les particules de types i et j, ils ont pour expression *cf.* Michaud & Proffit [1993] :

$$K_{ij} = \frac{16}{3} n_i n_j m_{ij} \Omega_{ij}^{(11)}. \tag{7.50}$$

7.8.7 Equations de Burgers pour les x_i et μ.

En remplaçant la charge de chaque ion par sa charge moyenne tenant compte de son taux d'ionisation, le poids moléculaire moyen est :

$$\mu^{-1} = \sum_{j\neq e}(1+\bar{Z}_j)\frac{X_j}{\nu_j} = \sum_{j\neq e}(1+\bar{Z}_j)x_j.$$

Pour les dérivations, on utilisera :

$$d\mu = -\mu^2 \sum_{j\neq e}(1+\bar{Z}_j)dx_j,\ \frac{\partial\mu}{\partial x_k} = -\mu^2(1+\bar{Z}_k),\ k \neq e.$$

Avec l'hypothèse du gaz parfait, pour chaque espèce i, les pressions totale et partielles s'écrivent :

$$P = \frac{\rho\mathcal{R}T}{\mu} + P_{\mathrm{R}} = \sum_i P_i,$$

$$P_i = n_i kT + P_{i\mathrm{R}} = \rho x_i N_0 kT + P_{i\mathrm{R}} = \frac{P\mu}{\mathcal{R}T} x_i N_0 kT + P_{i\mathrm{R}} = P\mu x_i + P_{i\mathrm{R}},$$

puisque la constante des gaz parfaits vérifie $\mathcal{R} \equiv kN_0$. On a donc :

$$n_i k = \frac{P}{T}\mu x_i.$$

En posant, pour les électrons :

$$x_e \equiv \frac{n_e}{\rho N_0} = \frac{\sum_{i\neq e}\bar{Z}_i n_i}{\rho N_0} = \sum_{i\neq e}\bar{Z}_i x_i,\ \frac{\partial x_e}{\partial x_k} = \bar{Z}_k,\ k \neq e,$$

comme $P_{\rm eR}=0$, on retrouve bien la pression électronique :

$$P\mu x_e=\frac{\rho\mathcal{R}T}{\mu}\mu\frac{n_e}{\rho N_0}=n_ekT=P_e.$$

L'hypothèse d'équilibre quasi-statique et $\rho_i=\rho\nu_ix_i$ (Eq. 7.49) entrainent :

$$\frac{\rho_i}{\rho}\frac{dP}{dr}=x_i\nu_i\rho g_{\rm t},$$

et :

$$n_ik\frac{dT}{dr}=\frac{P}{T}\mu x_i\frac{T}{P}\frac{\partial\ln T}{\partial\ln P}\frac{dP}{dr}=\mu x_i\nabla\rho g_{\rm t},$$

d'où :

$$\begin{aligned}\frac{dP_i}{dr}&=\frac{d}{dr}P\mu x_i+\frac{dP_{\rm iR}}{dr}=\mu x_i\frac{dP}{dr}+\frac{dP_{\rm iR}}{dr}+\mu P\frac{dx_i}{dr}+Px_i\frac{d\mu}{dr}\\&=x_i\rho\mu g_{\rm t}+\rho_ig_{{\rm R}i}+\mu P\frac{dx_i}{dr}-Px_i\mu^2\sum_{j\neq e}(1+\bar Z_j)\frac{dx_j}{dr}\\&=x_i\rho(\mu g_{\rm t}+\nu_ig_{{\rm R}i})+P\mu\left(\frac{dx_i}{dr}-x_i\mu\sum_{j\neq e}(1+\bar Z_j)\frac{dx_j}{dr}\right)\end{aligned}$$

On a par ailleurs :

$$n_i\bar Z_ieE=\rho N_0x_i\bar Z_ieE,\ \sum_i\nu_ix_iw_i=0,\ \sum_i\bar Z_ix_iw_i=0.$$

On introduit les relations précédentes dans les équations de Burgers Eq. 7.46 (Page 103), Eq. 7.47 (Page 103) et de conservation Eq. 7.48 (Page 103). Les coefficients de résistance K_{ij} sont de la forme :

$$K_{ij}=\frac{16}{3}n_in_jm_{ij}\Omega_{ij}^{(11)}=\frac{16}{3}(\rho N_0)^2x_ix_jm_{ij}\Omega_{ij}^{(11)}=k_{ij}x_ix_j\rho^2,\ k_{ij}\equiv\frac{16N_0^2}{3}m_{ij}\Omega_{ij}^{(11)}.$$

En fonction des x_i et de μ les équations de Burgers s'écrivent :

$$\begin{aligned}&x_i\rho\left(\mu g_{\rm t}-\nu_i\left(g_{\rm t}-g_{{\rm R}i}\right)\right)+P\mu\left(\frac{dx_i}{dr}-x_i\mu\sum_{j\neq e}(1+\bar Z_j)\frac{dx_j}{dr}\right)-\rho N_0x_i\bar Z_ieE\\&=\rho^2\left\{x_i\sum_jk_{ij}x_j(w_j-w_i)-x_i\sum_jk_{ij}x_jz_{ij}\frac{m_ir_j-m_jr_i}{m_i+m_j}\right\},\ i=1,\dots,c,\\&-x_i\rho\nabla\mu g_{\rm t}=\rho^2\left\{x_i\sum_jk_{ij}x_jz_{ij}\frac{m_j(w_j-w_i)}{m_i+m_j}+x_i\frac{4}{25}k_{ii}x_iz''_{ii}r_i\right.\\&+\frac{2}{5}r_ix_i\sum_{j\neq i}k_{ij}x_j\frac{3m_i^2+m_j^2z'_{ij}+\frac45m_im_jz''_{ij}}{(m_i+m_j)^2}\\&\left.-\frac{2}{5}x_i\sum_{j\neq i}k_{ij}x_jr_j\frac{m_im_j(3+z'_{ij}-\frac45z''_{ij})}{(m_i+m_j)^2}\right\},\ i=1,\dots,c,e,\end{aligned}$$

ou encore, en posant :

$$\nu_{i\mathrm{R}} \equiv \nu_i\left(1-\frac{g_{\mathrm{R}i}}{g_\mathrm{t}}\right),\ \frac{\partial\nu_{i\mathrm{R}}}{\partial x_i}=-\frac{\nu_i}{g_\mathrm{t}}\left(\frac{\partial g_{\mathrm{R}i}}{\partial x_j}-\frac{g_{\mathrm{R}i}}{g_\mathrm{t}}\frac{\partial g_\mathrm{t}}{\partial x_j}\right)$$

puis arrangements et division de chaque membre par $x_i\rho^2$:

$$\begin{aligned}\frac{g_\mathrm{t}}{\rho}(\mu-\nu_{i\mathrm{R}}) &+ \frac{\mathcal{R}T}{\rho}\left(\frac{1}{x_i}\frac{dx_i}{dr}-\mu\sum_{j\neq e}(1+\bar Z_j)\frac{dx_j}{dr}\right)=\sum_{j\neq i}k_{ij}x_j(w_j-w_i)\\ &- \sum_{j\neq i}k_{ij}x_jz_{ij}\frac{m_ir_j-m_jr_i}{m_i+m_j}+\frac{\bar Z_ieN_0E}{\rho},\ i=1,\dots,c, \qquad (7.51)\\ -\frac{\mu\nabla g_\mathrm{t}}{\rho} &= \sum_{j\neq i}k_{ij}x_jz_{ij}\frac{m_j(w_j-w_i)}{m_i+m_j}+\\ &+ r_i\left(\frac{4}{25}k_{ii}x_iz''_{ii}+\frac{2}{5}\sum_{j\neq i}k_{ij}x_j\frac{3m_i^2+m_j^2z'_{ij}+\frac45 m_im_jz''_{ij}}{(m_i+m_j)^2}\right)-\\ &- \frac25\sum_{j\neq i}k_{ij}x_jr_j\frac{m_im_j(3+z'_{ij}-\frac45 z''_{ij})}{(m_i+m_j)^2},\ i=1,\dots,c,e. \qquad (7.52)\end{aligned}$$

Les données sont la température T, la densité ρ, les gravités g_t et $g_{\mathrm{R}i}$, le gradient ∇, les abondances des ions $x_i,\ i=1,\dots,c$. Des abondances, on déduit le nombre d'électrons et le poids moléculaire moyen.

PB `Faute de mieux, le poids moléculaire moyen, les pressions partielles, ne peuvent pas être cohérents avec ceux utilisés dans l'équation d'état retenue par` CESAM2*k*.

Les inconnues sont les $c{+}1$ vitesses de diffusion $w_1,\dots,w_c,w_e$, les $c{+}1$ flux de chaleur $r_1,\dots,r_c,r_e$ et le champ électrique E, soit $2c+3$ inconnues. On ajoute aux $2c+1$ Eq. 7.51 (Page 106) et Eq. 7.52 (Page 106), les deux équations de conservation :

$$\sum_{i=1}^{c}\nu_ix_iw_i=0,\ \sum_{i=1}^{e}\bar Z_ix_iw_i=0. \qquad (7.53)$$

Les Eq. 7.51 (Page 106), Eq. 7.52 (Page 106) écrites pour les x_i et μ et Eq. 7.53 (Page 106) forment le système linéaire :

$$A\omega=\gamma+G\,D_x,\ G\equiv\frac{\mathcal{R}T}{\rho}\Delta,$$

on a noté :

$$\omega\equiv(w_1,\dots,w_c,w_e,r_1,\dots,r_c,r_e,E)^T,$$

$$D_x\equiv(\frac{dx_1}{dr},\dots,\frac{dx_c}{dr},0,\dots,0)^T,$$

$$\gamma\equiv\frac{g_\mathrm{t}}{\rho}\left(\mu-\nu_{1\mathrm{R}},\dots,\mu-\nu_{c\mathrm{R}},-\mu\nabla,\dots,-\mu\nabla,0,0\right)^T.$$

$$\frac{\partial}{\partial x_j}\left(\frac{\mu g_\mathrm{t}\nabla}{\rho}\right)=\frac{g_\mathrm{t}\nabla}{\rho}\frac{\partial\mu}{\partial x_j}+\frac{\mu\nabla}{\rho}\frac{\partial g_\mathrm{t}}{\partial x_j}+\left\{\frac{\mu g_\mathrm{t}}{\rho}\frac{\partial\nabla}{\partial x_1}-\frac{\mu g_\mathrm{t}\nabla}{\rho^2}\frac{\partial\rho}{\partial x_1}\right\}_{j=1},$$

$$\frac{\partial}{\partial x_j}\left(\frac{g_\mathrm{t}(\mu-\nu_\mathrm{iR})}{\rho}\right)=\frac{(\mu-\nu_\mathrm{iR})}{\rho}\frac{\partial g_\mathrm{t}}{\partial x_j}+\frac{g_\mathrm{t}}{\rho}\left(\frac{\partial \mu}{\partial x_j}-\frac{\partial \nu_\mathrm{iR}}{\partial x_j}\right)+\left\{\frac{g_\mathrm{t}}{\rho^2}(\mu-\nu_\mathrm{iR})\frac{\partial \rho}{\partial x_1}\right\}_{j=1}.$$

$$\Delta \equiv \mathrm{Diag}\left(\frac{1}{x_1},\frac{1}{x_2},\dots,\frac{1}{x_c},0,\dots,0\right)-\mu\mathcal{M}, \tag{7.54}$$

$$\mathcal{M} \equiv \begin{pmatrix} (1+\bar{Z}_1) & (1+\bar{Z}_2) & (1+\bar{Z}_3) & \dots & (1+\bar{Z}_c) & 0 & 0 & \dots & 0 & 0 \\ (1+\bar{Z}_1) & (1+\bar{Z}_2) & (1+\bar{Z}_3) & \dots & (1+\bar{Z}_c) & 0 & 0 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ (1+\bar{Z}_1) & (1+\bar{Z}_2) & (1+\bar{Z}_3) & \dots & (1+\bar{Z}_c) & 0 & 0 & \dots & 0 & 0 \\ (1+\bar{Z}_1) & (1+\bar{Z}_2) & (1+\bar{Z}_3) & \dots & (1+\bar{Z}_c) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix} \tag{7.55}$$

La structure de la matrice A, de dimension $[2c+3 \times 2c+3]$, est la suivante :

$$\begin{pmatrix} A_w & A_r & A_e \\ A_c & 0 & 0 \end{pmatrix},$$

les dimensions des sous matrices sont respectivement $A_w[2c+1\times c+1]$, $A_r[2c+1\times c+1]$, $A_e[2c+1\times 1]$ et $A_c[2\times c+1]$. En notant :

$$p_{ij}\equiv k_{ij}x_j,\ \ q_{ij}\equiv \frac{k_{ij}x_jz_{ij}m_j}{m_i+m_j}=p_{ij}\frac{z_{ij}m_j}{m_i+m_j},$$

$$A_w \equiv \begin{pmatrix} -\sum_{j\neq 1}p_{1j} & p_{12} & p_{13} & \dots & p_{1c} & p_{1e} \\ p_{21} & -\sum_{j\neq 2}p_{2j} & p_{23} & \dots & p_{2c} & p_{2e} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ p_{c1} & p_{c2} & p_{c3} & \dots & -\sum_{j\neq c}p_{cj} & p_{ce} \\ -\sum_{j\neq 1}q_{1j} & q_{12} & q_{13} & \dots & q_{1c} & q_{1e} \\ q_{21} & -\sum_{j\neq 2}q_{2j} & q_{23} & \dots & q_{2c} & q_{2e} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ q_{c1} & q_{c2} & q_{c3} & \dots & -\sum_{j\neq c}q_{cj} & q_{ce} \end{pmatrix}, \tag{7.56}$$

avec les notations :

$$\bar{q_{ij}} \equiv -\frac{k_{ij}x_jz_{ij}m_i}{m_i+m_j}=-q_{ij}\frac{m_i}{m_j}, \tag{7.57}$$

$$d_i \equiv \frac{4}{25}k_{ii}x_iz''_{ii}+\frac{2}{5}\sum_{j\neq i}k_{ij}x_j\frac{3m_i^2+m_j^2z'_{ij}+\frac{4}{5}m_im_jz''_{ij}}{(m_i+m_j)^2}, \tag{7.58}$$

$$f_{ij} \equiv -\frac{2}{5}k_{ij}x_j\frac{m_im_j(3+z'_{ij}-\frac{4}{5}z''_{ij})}{(m_i+m_j)^2}, \tag{7.59}$$

$$A_r \equiv \begin{pmatrix} \sum_{j\neq 1} q_{1j} & \bar{q_{12}} & \bar{q_{13}} & \dots & \bar{q_{1c}} & \bar{q_{1e}} \\ \bar{q_{21}} & \sum_{j\neq 2} q_{2j} & \bar{q_{23}} & \dots & \bar{q_{2c}} & \bar{q_{2e}} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \bar{q_{c1}} & \bar{q_{c2}} & \bar{q_{c3}} & \dots & \sum_{j\neq c} q_{cj} & \bar{q_{ce}} \\ d_1 & f_{12} & f_{13} & \dots & f_{1c} & f_{1e} \\ f_{21} & d_2 & f_{23} & \dots & f_{2c} & f_{2e} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ f_{c1} & f_{c2} & f_{c3} & \dots & d_c & f_{ce} \\ f_{e1} & f_{e2} & f_{e3} & \dots & f_{ec} & d_e \end{pmatrix}, \tag{7.60}$$

$$A_e \equiv \frac{N_0 e}{\rho}(\bar{Z}_1, \dots, \bar{Z}_c, 0, \dots, 0)^T, \tag{7.61}$$

$$A_c \equiv \begin{pmatrix} \nu_1 x_1 & \nu_2 x_2 & \dots & \nu_c x_c & 0 & 0 & \dots & 0 \\ \bar{Z}_1 x_1 & \bar{Z}_2 x_2 & \dots & \bar{Z}_c x_c & \bar{Z}_e x_e & 0 & \dots & 0 \end{pmatrix}. \tag{7.62}$$

La solution s'écrit :

$$\omega = A^{-1}\gamma + A^{-1}G\, D_x = v + B\, D_x \Rightarrow, \tag{7.63}$$

ou encore, pour les ions, $i = 1, \dots, c$:

$$w_i = v_i + \sum_{j=1}^{c} b_{ij}\frac{dx_j}{dr},\ i = 1, \dots, c,$$

les v_i sont les c premiers coefficients de :

$$v = A^{-1}\gamma, \tag{7.64}$$

les $b_{ij},\ (i, j = 1, \dots, c)$, sont les coefficients des c premières lignes et colonnes de

$$B = A^{-1}G. \tag{7.65}$$

Pour l'ion i l'équation de diffusion devient :

$$\begin{aligned} \frac{\partial n_i}{\partial t} &= -\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 n_i w_i) + \left(\frac{\partial n_i}{\partial t}\right)_{\text{nucl.}} = \\ \frac{\partial \rho N_0 x_i}{\partial t} &= -\frac{1}{r^2}\frac{\partial}{\partial r}\left\{N_0 r^2 \rho x_i \left(\sum_{j=1}^{c} b_{ij}\frac{\partial x_j}{\partial r} + v_i\right)\right\} + \left(\frac{\partial \rho N_0 x_i}{\partial t}\right)_{\text{nucl.}} \end{aligned}$$

ou encore, en notant $\frac{D\star}{Dt} \equiv \frac{\partial \star}{\partial t} + \mathcal{V} * \nabla\star$ l'opérateur de dérivation lagrangien "en suivant le mouvement" :

$$\frac{Dx_i}{Dt} = -\frac{4\pi}{4\pi r^2\rho}\frac{\partial}{\partial r}\left\{\frac{4\pi r^4\rho^2}{4\pi r^2\rho}x_i\left(\sum_{j=1}^{c} b_{ij}\frac{\partial x_j}{\partial r} + v_i\right)\right\} + \left(\frac{Dx_i}{Dt}\right)_{\text{nucl.}} \Longrightarrow$$

$$\frac{Dx_i}{Dt} = -\frac{\partial}{\partial m}\left\{4\pi r^2\rho\left(4\pi r^2\rho x_i\sum_{j=1}^{c} b_{ij}\frac{\partial x_j}{\partial m} + v_i x_i\right)\right\} + \left(\frac{Dx_i}{Dt}\right)_{\text{nucl.}}.$$

On a utilisé l'équation de continuité :

$$0 = \frac{\partial \rho}{\partial t} + \nabla * \rho \mathcal{V} = \frac{\partial \rho}{\partial t} + \rho \nabla * \mathcal{V} + \mathcal{V} * \nabla \rho = x_i \frac{\partial \rho}{\partial t} + x_i \rho \nabla * \mathcal{V} + x_i \mathcal{V} * \nabla \rho,$$

et la conservation des particules d'espèce i :

$$\nabla * x_i \rho \mathcal{V} = 0 \Rightarrow \rho \mathcal{V} * \nabla x_i = -x_i \rho \nabla * \mathcal{V} - x_i \mathcal{V} * \nabla \rho,$$

$$\Rightarrow \frac{\partial \rho x_i}{\partial t} = x_i \frac{\partial \rho}{\partial t} + \rho \frac{\partial x_i}{\partial t} - x_i \frac{\partial \rho}{\partial t} - x_i \rho \nabla * \mathcal{V} - x_i \mathcal{V} * \nabla \rho = \rho \left(\frac{\partial x_i}{\partial t} + \mathcal{V} * \nabla x_i \right) = \rho \frac{D x_i}{D t}.$$

En posant :

$$D_i \equiv (d_{i1}, \ldots, d_{ic},) \equiv -(x_i b_{i1}, \ldots, x_i b_{ic},)\,, \; i = 1, \ldots, c, \tag{7.66}$$

on obtient la forme de l'équation de diffusion utilisée dans CESAM2k[18] :

$$\frac{D x_i}{D t} = -\frac{\partial F_i}{\partial m} + \Psi_i, \; F_i = -4\pi R^2 \rho \left(4\pi R^2 \rho D_i * \nabla_{\mathrm{m}} \mathcal{X} - v_i x_i \right)$$

où $\nabla_{\mathrm{m}} \mathcal{X}$ est le vecteur :

$$\left(\frac{\partial x_1}{\partial m}, \ldots, \frac{\partial x_{\mathrm{c}}}{\partial m} \right)^{\mathrm{T}}.$$

7.8.8 Jacobien.

CESAM2k intégrant l'équation de diffusion avec un schéma implicite, il est nécessaire de calculer les dérivées :

$$\frac{\partial v_i}{\partial x_k}, \; \frac{\partial d_{ij}}{\partial x_k} = -x_i \frac{\partial b_{ij}}{\partial x_k} - b_{ij} \delta_{ik}, \; i, j, k = 1, \ldots, c, \; \delta_{ik} = 1 \text{ si } i = k, \; 0 \text{ sinon.} \tag{7.67}$$

En dérivant :

$$A^{-1} A = I \Rightarrow \frac{\partial A^{-1}}{\partial x_k} A + A^{-1} \frac{\partial A}{\partial x_k} = 0 \Rightarrow \frac{\partial A^{-1}}{\partial x_k} = -A^{-1} \frac{\partial A}{\partial x_k} A^{-1},$$

d'où :

$$\frac{\partial A^{-1} \gamma}{\partial x_k} = \frac{\partial A^{-1}}{\partial x_k} \gamma + A^{-1} \frac{\partial \gamma}{\partial x_k} = -A^{-1} \frac{\partial A}{\partial x_k} A^{-1} \gamma + A^{-1} \frac{\partial \gamma}{\partial x_k} = A^{-1} \left(\frac{\partial \gamma}{\partial x_k} - \frac{\partial A}{\partial x_k} v \right),$$

et de façon similaire :

$$\frac{\partial B}{\partial x_k} = A^{-1} \left(\frac{\partial G}{\partial x_k} - \frac{\partial A}{\partial x_k} B \right). \tag{7.68}$$

[18] Avec la variable d'espace Lagrangienne m les opérateurs $\frac{D}{Dt}$ et $\frac{\partial}{\partial t}$ sont équivalents.

7.8.9 Accélérations radiatives

L'accélération radiative ressentie par les isotopes résulte de l'impulsion des photons qu'ils absorbent. En raison de l'anisotropie du rayonnement, même dans le milieu épais, pour la plupart des isotopes, la composante radiale de l'accélération radiative diminue la gravité locale et donc la sédimentation. Deux descriptions dues à G.Alécian sont implantées dans CESAM2k.

On simplifie en considérant chaque isotope dans l'ensemble de ses états d'ionisation. m étant la masse à l'intérieur de la sphère de rayon r, la gravité effective $g_{\rm e}$ qui compense le gradient de pression dans l'équilibre quasi-statique, est somme de la partie purement gravitationnelle $g_{\rm G} = -Gm/r^2$, de l'accélération centrifuge moyenne $g_\Omega = \frac{2}{3}r\Omega^2$ et d'une partie radiative $g_{\rm R}$. On postule que $g_{\rm R}$ est la somme, pondérée par les masses partielles des particules ($\sum_{i\neq e} n_i m_i$) des accélérations radiatives $g_{{\rm R}i}$:

$$g_{\rm R} = \frac{\sum_{i\neq e} n_i m_i g_{{\rm R}i}}{\sum_{i\neq e} n_i m_i} = \frac{\sum_{i\neq e} \rho N_0 X_i g_{{\rm R}i}}{\sum_{i\neq e} \rho N_0 X_i} = \frac{\sum_{i\neq e} X_i g_{{\rm R}i}}{\sum_{i\neq e} X_i} = \sum_{i\neq e} x_i \nu_i g_{{\rm R}i}, \tag{7.69}$$

Sur la particule $i \neq e$, l'accélération g_i est somme de la gravité effective $g_{\rm e} \equiv g_{\rm G} + g_\Omega$ et de la partie radiative $g_{{\rm R}i}$:

$$g_i = g_{\rm e} + g_{{\rm R}i}.$$

Pour faciliter les algorithmes on pose, pour les électrons, $g_{{\rm R}e} \equiv 0$. On notera la gravité totale par :

$$g_{\rm t} \equiv g_{\rm e} + g_i = g_{\rm e} + \sum_i x_i \nu_i g_{{\rm R}i},\ \frac{\partial g_{\rm t}}{\partial x_i} = \nu_j g_{{\rm R}j} + \sum_i x_i \nu_i \frac{\partial g_{{\rm R}i}}{\partial x_j}.$$

La première description du calcul des accélérations radiatives[19] est effectuée dans le cadre de l'approximation proposée par Alécian [1985] et Alécian & Artru [1990]. Sont seuls pris en compte les effets des transitions lié-lié. C'est une approximation semi analytique où la dépendance de l'accélération radiative à la concentration apparaît dans un terme séparé de celui contenant les propriétés atomiques des ions. Cette approximation permet donc de ne calculer qu'un nombre réduit de quantités lorsque la concentration locale évolue au cours du temps. Pour l'élément $i \neq e$ l'accélération radiative est obtenue par une somme pondérée des accélérations radiatives de chaque ion j :

$$g_{{\rm R}i} = \sum_{j=0}^{Z_i} x_{j,i} g_{{\rm R}i,j},\ g_{{\rm R}i,j} = g_{{\rm R}_0 i,j}\sqrt{1 + \frac{C_{j,i}}{C_{{\rm S}j,i}}},\ C_{{\rm S}j,i} = b\Psi^2_{j,i},\ g_{{\rm R}_0 i,j} = q_i \Phi_{j,i},$$

où $C_{j,i} = n_{j,i}/n_{\rm H}$ est la concentration, par rapport à l'hydrogène, de l'ion j de l'élément chimique i, $C_{{\rm S}j,i}$ sa concentration de saturation et $g_{{\rm R}_0 i,j}$ l'accélération radiative à la limite de concentration nulle. On remarquera que :

$$\frac{C_{j,i}}{C_{{\rm S}j,i}} = \frac{x_{j,i}}{x_{{\rm S}j,i}}.$$

Les quantités b et q_i sont données par :

$$b = \frac{m_e m_p \mathcal{C}}{2e^2 \nu_{\rm H}} \frac{n_e \kappa}{\sqrt{T}},\ q_i = \frac{\pi k^3 e^2}{2a\,\mathcal{C}^5 m_e m_u} \frac{l}{Tr^2 \nu_i},$$

[19] Ce paragraphe a été rédigé en grande partie par G. Alécian.

$\mathcal{C}$ est la célérité de la lumière, m_p la masse du proton, κ l'opacité moyenne de Rosseland, h la constante de Plank, a la constante de la radiation, k la constante de Boltzman, r le rayon local et l la luminosité locale. Les quantités $\Psi^2_{j,i}$ et $\Phi_{j,i}$ sont tabulées dans le cas de l'approximation #2 décrite dans Alécian *et al.* [1990] et ce, pour des masses stellaires de $1M_\odot$ à $2M_\odot$. Les données de la base Topbase sont utilisées. Ces tables, qui correspondent à des masses différentes, sont lues au premier appel de la routine `alecian1` décrite au § 8.5 (Page 144). Les transitions radiatives pour les isotopes d'un même élément se produisant à des fréquences très proches, l'opacité monochromatique de l'ensemble des isotopes d'un même élément est pratiquement égale à la somme des opacités monochromatiques de ses divers isotopes. Dans CESAM2k, les isotopes apparaissent comme des éléments séparés. Pour le calcul des accélérations radiatives, les isotopes d'un même élément sont identifiés et leurs abondances sont additionnées de façon à traiter correctement l'effet de saturation des transitions lié-lié. Les accélérations radiatives sont ensuite restituées isotope par isotope.
Pour la description de la physique du second formalisme du calcul des accélérations radiatives, se référer aux publications de G.Alécian.

7.9 Diffusion du moment cinétique

Le formalisme de la diffusion du moment cinétique implanté dans CESAM2k est celui développé par Talon *et al.* [1997]. Les coefficients de diffusion peuvent être calculés suivant les prescriptions de Palacios *et al.* [2003] ou de Matis, Palacios & Zahn [2004].

La diffusion du moment cinétique et celle des espèces chimiques sont couplées par la turbulence du milieu, principalement au voisinage des limites zone radiative / zone convective. La résolution numérique du système d'équations régissant ces deux processus de diffusion s'est avérée extrèmement instable. La solution retenue a été de résoudre séparément les sytèmes d'équations correspondant respectivement au moment cinétique et aux espèces chimiques. La méthode des éléments finis utilisée pour résoudre les équations de la diffusion du moment cinétique ne différant de celle utilisée pour la diffusion des éléments chimiques que par la prise en compte des conditions limites.

7.9.1 Les variables

- $x_i,\ i = 1, \ldots, n_c$: abondances par mole
- $y_1 \equiv \Omega$: vitesse angulaire s^{-1},
- $y_2 \equiv U$: vitesse de circulation méridienne,
- $y_3 \equiv \theta = \frac{\tilde\rho}{\rho}$: taux de fluctuation de la densité le long d'une isobare
- $y_4 \equiv \Lambda = \frac{\tilde\mu}{\mu}$: taux de fluctuation du poids moléculaire moyen le long d'une isobare
- $y_5 \equiv \psi \equiv \frac{\tilde T}{T} = \frac{\varphi\Lambda-\theta}{\delta}$: taux de fluctuation de la température le long d'une isobare

7.9.2 Changement de variable $M \to \nu = \left(\frac{M}{M_\odot}\right)^{\frac{2}{3}}$

Les abondances et la vitesse angulaire étant des variables lagrangiennes, pour l'intégration des équations d'évolution de la composition chimique et du moment cinétique, on utilise la variable d'espace ν. Les transformations eulérien $\leftrightarrow$ lagrangien sont les suivantes :

$$M = M_\odot m \ ; \ \nu = m^{\frac{2}{3}} \ ; \ m = \nu^{\frac{3}{2}} \ ; \ M = M_\odot m \ ; \ d\nu = \frac{2}{3} m^{-\frac{1}{3}} dm \ ; \ dm = \frac{3}{2}\sqrt{\nu} d\nu$$

$$dM = M_\odot dm = M_\odot \frac{3}{2}\sqrt{\nu} d\nu \ ; \ \frac{\partial}{\partial M} = \frac{2}{3 M_\odot \sqrt{\nu}} \frac{\partial}{\partial \nu}$$

$$R = R_\odot r \ ; \ dM = \frac{3 M_\odot}{2}\sqrt{\nu} d\nu = 4\pi R^2 \rho dR = 4\pi R_\odot^2 r^2 \rho dR$$

$$dR = \frac{3 M_\odot}{8\pi R_\odot^2} \frac{\sqrt{\nu}}{r^2 \rho} d\nu \ ; \ \frac{\partial}{\partial R} = \frac{8\pi R_\odot^2 r^2 \rho}{3 M_\odot \sqrt{\nu}} \frac{\partial}{\partial \nu}$$

7.9.3 Quelques notations

- $l = \frac{L}{L_\odot}$: luminosité,
- ν_i : masse atomique de l'élément chimique i,
- $\bar{z}_i$: charge moyenne de l'élément chimique i en tenant compte de l'ionisation,
- $\dot{\mathcal{M}}$: taux de perte de masse par unité de temps,
- $\dot{\mathcal{M}}_\Omega \leq 0$: taux de perte de moment cinétique par unité de masse et de temps.

Dans les algorithmes, les variables relatives à la rotation ont un nom ou une extension comprenant `rot, rota` ou encore `w`.

7.9.4 Expressions de H_P, H_T, ∇_μ, χ etc... et dérivées

$$H_\mathrm{P} = -\frac{\partial R}{\partial \ln P} = \frac{P}{\rho g} = \frac{P R^2}{G M \rho} = \frac{R_\odot^2 r^2 P}{G M_\odot \nu^{\frac{3}{2}} \rho}$$

$$\nabla = \frac{\partial \ln T}{\partial \ln P}$$

$$H_\mathrm{T} = -\frac{\partial R}{\partial \ln T} = -\frac{\partial R}{\partial \ln P}\frac{\partial \ln P}{\partial \ln T} = \frac{H_\mathrm{P}}{\nabla}$$

$$x_i = \frac{X_i}{\nu_i}$$

$$\mu^{-1} = \sum_i (1 + \bar{z}_i) x_i$$

$$\mu^{-1} \simeq \frac{1}{16}(20X + 12 - 3Z) = \frac{1}{16}(20X + 12 - 3(1 - X - Y)) = \frac{1}{16}(23X + 3Y + 9) = \frac{1}{16}(23\nu_\mathrm{H} x_\mathrm{H} + 3\nu_\mathrm{He} x_\mathrm{He} + 9)$$

$$\frac{\partial \mu}{\partial x_i} = -\mu^2 (1 + \bar{z}_i)$$

$$\frac{\partial \ln \mu}{\partial \nu} = \frac{1}{\mu}\sum_i \frac{\partial \mu}{\partial x_i}\frac{\partial x_i}{\partial \nu} = -\mu \sum_i (1+\bar{z}_i)\frac{\partial x_i}{\partial \nu}$$

$$\frac{\partial \ln \mu}{\partial \nu} \simeq -\frac{\mu}{16}\left(23\nu_{\mathrm{H}}\frac{\partial x_{\mathrm{H}}}{\partial \nu} + 3\nu_{\mathrm{He}}\frac{\partial x_{\mathrm{He}}}{\partial \nu}\right)$$

$$\varepsilon_\mu = \frac{\partial \ln \varepsilon}{\partial \ln \mu} = \frac{\mu}{\varepsilon}\sum_i \frac{\partial \varepsilon}{\partial x_i}\frac{\partial x_i}{\partial \mu} = -\frac{1}{\mu\varepsilon}\sum_i \frac{1}{1+\bar{z}_i}\frac{\partial \varepsilon}{\partial x_i}$$

$$\frac{\partial \ln \mu}{\partial X} \simeq \frac{1}{\mu}\frac{\partial \mu}{\partial X} = -\frac{23\mu}{16},\ \frac{\partial \ln \mu}{\partial Y} \simeq -\frac{3\mu}{16}$$

$$\varepsilon_\mu \simeq \frac{1}{\varepsilon}\left(\frac{\partial \varepsilon}{\partial X}\frac{\partial X}{\partial \ln \mu} + \frac{\partial \varepsilon}{\partial Y}\frac{\partial Y}{\partial \ln \mu}\right) = -\frac{16}{\varepsilon\mu}\left(\frac{1}{23\nu_{\mathrm{H}}}\frac{\partial \varepsilon}{\partial x_{\mathrm{H}}} + \frac{1}{3\nu_{\mathrm{He}}}\frac{\partial \varepsilon}{\partial x_{\mathrm{He}}}\right)$$

$$\varepsilon_{\mathrm{T}} = \frac{\partial \ln \varepsilon}{\partial \ln T} = \frac{T}{\varepsilon}\frac{\partial \varepsilon}{\partial T}$$

$$\varphi = \frac{\partial \ln \rho}{\partial \ln \mu} = \frac{\mu}{\rho}\frac{\partial \rho}{\partial \mu} = \frac{\mu}{\rho}\sum_i \frac{\partial \rho}{\partial X_i}\frac{\partial X_i}{\partial \mu} = -\frac{1}{\mu\rho}\sum_i \frac{\nu_i}{1+\bar{z}_i}\frac{\partial \rho}{\partial X_i}$$

$$\frac{\partial \ln \rho}{\partial \nu} = \alpha\frac{\partial \ln P}{\partial \nu} - \delta\frac{\partial \ln T}{\partial \nu} + \varphi\frac{\partial \ln \mu}{\partial \nu} \Longrightarrow \quad \text{si } \frac{\partial \ln \mu}{\partial \nu} \neq 0$$

$$\varphi = \left(\frac{\partial \ln \rho}{\partial \nu} - \alpha\frac{\partial \ln P}{\partial \nu} + \delta\frac{\partial \ln T}{\partial \nu}\right)\Big/\left(\frac{\partial \ln \mu}{\partial \nu}\right)$$

$$\nabla_\mu = \frac{\partial \ln \mu}{\partial \ln P} = \frac{\partial \ln \mu}{\partial R}\frac{\partial R}{\partial \ln P} = -\frac{\partial \ln \mu}{\partial R}H_{\mathrm{P}} = -\frac{8\pi R_\odot^2 r^2\rho}{3M_\odot\sqrt{\nu}}H_{\mathrm{P}}\frac{\partial \ln \mu}{\partial \nu} =$$
$$\frac{8\pi R_\odot^2}{3M_\odot}\frac{r^2\rho\mu H_{\mathrm{P}}}{\sqrt{\nu}}\sum_i (1+\bar{z}_i)\frac{\partial x_i}{\partial \nu}$$

$$\nabla_\mu \simeq \frac{\pi R_\odot^2 r^2\rho}{6M_\odot\sqrt{\nu}}H_{\mathrm{P}}\left(23\nu_{\mathrm{H}}\frac{\partial x_{\mathrm{H}}}{\partial \nu} + 3\nu_{\mathrm{He}}\frac{\partial x_{\mathrm{He}}}{\partial \nu}\right)$$

$$\nabla_{\mathrm{rad}} = \frac{3\kappa LP}{16\pi acGMT^4} = \frac{3L_\odot}{16\pi acGM_\odot}\frac{\kappa lP}{\nu^{\frac{3}{2}}T^4}$$

$$g = \frac{GM}{R^2} = \frac{GM_\odot}{R_\odot^2}\frac{\nu^{\frac{3}{2}}}{r^2},\ \frac{\tilde{g}}{g} = \frac{4R^3}{3GM}\Omega^2 = \frac{4R_\odot^3}{3GM_\odot}\frac{r^3}{\nu^{\frac{3}{2}}}\Omega^2$$

$$N_T^2 = \frac{g\delta}{H_{\mathrm{P}}}(\nabla_{\mathrm{ad}} - \nabla),\ N_\mu^2 = \frac{g\varphi}{H_{\mathrm{P}}}\nabla_\mu$$

$$\chi = \frac{4acT^3}{3\kappa\rho}$$

$$\chi_{\mathrm{T}} = \frac{\partial \ln \chi}{\partial \ln T} = 3 - \frac{\partial \ln \kappa}{\partial \ln T} - \frac{\partial \ln \rho}{\partial \ln T} = 3 - \frac{T}{\kappa}\frac{\partial \kappa}{\partial T} + \delta$$

$$\chi_\mu = \frac{\partial \ln \chi}{\partial \ln \mu} = -\frac{\partial \ln \kappa}{\partial \ln \mu} - \frac{\partial \ln \rho}{\partial \ln \mu} = -\frac{\mu}{\kappa}\sum_i \frac{\partial \kappa}{\partial x_i}\frac{\partial x_i}{\partial \mu} - \varphi = -\frac{1}{\mu\kappa}\sum_i \frac{\partial \kappa}{\partial X_i}\frac{\nu_i}{1+\bar{z}_i} - \varphi$$

$$K = \frac{\chi}{\rho c_{\mathrm{P}}} = \frac{4acT^3}{3\kappa\rho^2 c_{\mathrm{P}}}$$

$$\frac{\bar{\varepsilon} + \bar{\varepsilon}_g}{\varepsilon_{\mathrm{m}}} = \frac{M}{L}\frac{\partial L}{\partial M} = \frac{\partial \ln L}{\partial \ln M} = \frac{\partial \ln l}{\partial \ln m}$$

$$f_\varepsilon = \frac{\bar{\varepsilon}}{\bar{\varepsilon} + \bar{\varepsilon}_g} = \frac{\partial M}{\partial L}\varepsilon_{nuc} = \frac{M_\odot}{L_\odot}\varepsilon_{nuc}\frac{\partial m}{\partial l}$$

$$f_\varepsilon \frac{\bar{\varepsilon}+\bar{\varepsilon}_g}{\varepsilon_\mathrm{m}} = \frac{\bar{\varepsilon}}{\bar{\varepsilon}+\bar{\varepsilon}_g}\frac{\bar{\varepsilon}+\bar{\varepsilon}_g}{\varepsilon_\mathrm{m}} = \frac{\bar{\varepsilon}}{\varepsilon_\mathrm{m}} = \frac{M_\odot}{L_\odot}\frac{\varepsilon_{nuc}}{l}\nu^{\frac{3}{2}}$$
$$\rho_m = \frac{3M}{4\pi R^3} = \frac{3M_\odot}{4\pi R_\odot^3}\frac{\nu^{\frac{3}{2}}}{r^3}$$

7.9.5 Expressions de D_h, D_v et D_eff

Les coefficients de diffusivité horizontale D_h et verticale D_v sont donnés par Matis, Palacios & Zahn [2004], équations 7-10-19 :

$$D_\mathrm{h} = R\sqrt{\beta\Omega R\,|2V-\alpha U|},\ \beta = 1.5\,10^{-6},\ V = \frac{1}{6\rho R}\frac{\partial R^2\rho U}{\partial R},\ \alpha = \frac{1}{2}\frac{\partial \ln R^2\Omega}{\partial \ln R}$$
$$D_\mathrm{h}^2 = R^3\beta\Omega\left|\frac{1}{3R\rho}\frac{\partial R^2\rho U}{\partial R} - \frac{RU}{2R^2\Omega}\frac{\partial R^2\Omega}{\partial R}\right| = R^2\beta\Omega\left|\frac{1}{3\rho}\frac{\partial R^2\rho U}{\partial R} - \frac{U}{2\Omega}\frac{\partial R^2\Omega}{\partial R}\right| =$$
$$R^2\beta\Omega\left|\frac{R^2\rho}{3\rho}\frac{\partial U}{\partial R} + \frac{2R\rho U}{3\rho} + \frac{R^2U}{3\rho}\frac{\partial\rho}{\partial R} - \frac{2UR\Omega}{2\Omega} - \frac{UR^2}{2\Omega}\frac{\partial\Omega}{\partial R}\right| =$$
$$R^2\beta\Omega\left|\frac{R^2}{3}\frac{\partial U}{\partial R} + \frac{2}{3}UR + \frac{R^2U}{3\rho}\frac{\partial\rho}{\partial R} - UR - \frac{UR^2}{2\Omega}\frac{\partial\Omega}{\partial R}\right| =$$
$$R^4\beta\Omega\left|\frac{1}{3}\frac{\partial U}{\partial R} - \frac{U}{3R} + \frac{U}{3\rho}\frac{\partial\rho}{\partial R} - \frac{U}{2\Omega}\frac{\partial\Omega}{\partial R}\right| \Rightarrow$$
$$D_\mathrm{h}^2 = \frac{R_\odot^4\beta}{3}r^4\Omega\left|\frac{8\pi R_\odot^2}{3M_\odot}\frac{r^2\rho}{\sqrt{\nu}}\left[\frac{\partial U}{\partial\nu} + U\left(\frac{\partial\ln\rho}{\partial\nu} - \frac{3}{2\Omega}\frac{\partial\Omega}{\partial\nu}\right)\right] - \frac{U}{R_\odot r}\right|$$
$$D_\mathrm{h}^2 = C_1\Omega\frac{\partial U}{\partial\nu} + C_2\Omega U - C_3U\frac{\partial\Omega}{\partial\nu},\quad C_1 = \frac{8\pi R_\odot^6\beta}{9M_\odot}\frac{r^6\rho}{\sqrt{\nu}}$$
$$C_2 = C_1^* - C_2^*,\quad C_1^* = \frac{8\pi R_\odot^6\beta}{9M_\odot}\frac{r^6\rho}{\sqrt{\nu}}\frac{\partial\ln\rho}{\partial\nu},\quad C_2^* = \frac{R_\odot^3\beta}{3}r^3,\quad C_3 = \frac{4\pi R_\odot^6\beta}{3M_\odot}\frac{r^6\rho}{\sqrt{\nu}}$$

$$D_\mathrm{v} = \frac{Ri_c(K+D_\mathrm{h})R^2}{N_T^2 + N_\mu^2(1+K/D_\mathrm{h})}\left(\frac{\partial\Omega}{\partial R}\right)^2,\ Ri_c = \frac{1}{6}$$
$$D_\mathrm{v} = \frac{64\pi^2 Ri_c R_\odot^6}{9M_\odot^2}\frac{r^6\rho^2}{\nu}\frac{D_\mathrm{h}(1+\frac{K}{D_\mathrm{h}})}{\left[N_T^2 + N_\mu^2\left(1+\frac{K}{D_\mathrm{h}}\right)\right]}\left(\frac{\partial\Omega}{\partial\nu}\right)^2$$
$$D_\mathrm{eff} = \frac{R_\odot^2 r^2 U^2}{30D_\mathrm{h}} + D_\mathrm{v}$$

7.9.6 Transport du moment cinétique

Dans une zone radiative, le moment cinétique par unité de masse, $R^2\Omega = R^2y_1$, vérifie l'équation de transport (Matis & Zahn [2004], Equation B1) :

$$\rho\frac{DR^2\Omega}{Dt} = \frac{1}{R^2}\frac{\partial}{\partial R}\left[R^4\rho\left(\frac{\Omega U}{5} + D_\mathrm{v}\frac{\partial\Omega}{\partial R}\right)\right] - \rho\dot{\mathcal{M}}_\Omega \Rightarrow \tag{7.70}$$
$$R_\odot^2\frac{Dr^2\Omega}{Dt} = \frac{8\pi R_\odot^6 r^2\rho}{3M_\odot R_\odot^2 r^2\rho\sqrt{\nu}}\frac{\partial}{\partial\nu}\left[r^4\rho\left(\frac{\Omega U}{5} + \frac{8\pi R_\odot^2 r^2\rho D_\mathrm{v}}{3M_\odot\sqrt{\nu}}\frac{\partial\Omega}{\partial\nu}\right)\right] - \dot{\mathcal{M}}_\Omega \Rightarrow$$

$$\frac{Dr^2\Omega}{Dt} = \frac{8\pi R_\odot^2}{3M_\odot\sqrt{\nu}}\frac{\partial}{\partial\nu}\left[r^4\rho\left(\frac{\Omega U}{5} + \frac{8\pi R_\odot^2 r^2\rho}{3M_\odot\sqrt{\nu}}D_\mathrm{v}\frac{\partial\Omega}{\partial\nu}\right)\right] - \frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2} \Rightarrow$$
$$\sqrt{\nu}\frac{Dr^2\Omega}{Dt} = \frac{8\pi R_\odot^2}{15M_\odot}\frac{\partial}{\partial\nu}(r^4\rho\Omega U) + \frac{64\pi^2 R_\odot^4}{9M_\odot^2}\frac{\partial}{\partial\nu}\left(\frac{r^6\rho^2 D_\mathrm{v}}{\sqrt{\nu}}\frac{\partial\Omega}{\partial\nu}\right) - \frac{\dot{\mathcal{M}}_\Omega\sqrt{\nu}}{R_\odot^2}$$

L'intégration par éléments finis prend la forme :

$$\begin{aligned}
&< \sqrt{\nu}\left(\frac{r^2\Omega}{\Delta t} - \frac{r^2(t)\Omega(t)}{\Delta t} + \frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2}\right)\bullet S_j^{m_c} > \\
&+ < \left(\frac{8\pi R_\odot^2}{15M_\odot}r^4\rho\Omega U + \frac{64\pi^2 R_\odot^4}{9M_\odot^2}\frac{r^6\rho^2 D_\mathrm{v}}{\sqrt{\nu}}\frac{\partial\Omega}{\partial\nu}\right)\bullet\frac{\partial S_j^{m_c}}{\partial\nu} > \\
&- \left|\left(\frac{8\pi R_\odot^2}{15M_\odot}r^4\rho\Omega U + \frac{64\pi^2 R_\odot^4}{9M_\odot^2}\frac{r^6\rho^2 D_\mathrm{v}}{\sqrt{\nu}}\frac{\partial\Omega}{\partial\nu}\right)S_j^{m_c}\right|_{\nu_\mathrm{c}}^{\nu_\mathrm{s}} = 0
\end{aligned}$$

Avec les notations des algorithmes, les équations s'écrivent :

$$\begin{aligned}
&< (C_1\Omega - C_2)\bullet S_j^{m_c} > + < \left(C_3\Omega U + C_4\frac{\partial\Omega}{\partial\nu}\right)\bullet\frac{\partial S_j^{m_c}}{\partial\nu} > \\
&\qquad - \left|\left(C_3 U\Omega + C_4\frac{\partial\Omega}{\partial\nu}\right)S_j^{m_c}\right|_{\nu_\mathrm{c}}^{\nu_\mathrm{s}} = 0
\end{aligned} \tag{7.71}$$
$$C_1 = \frac{r^2\sqrt{\nu}}{\Delta t},\ C_2 = (C_{11}^* - C_{12}^*)\sqrt{\nu},\ C_{11}^* = \frac{r^2(t)\Omega(t)}{\Delta t},\ C_{12}^* = \frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2}$$
$$C_3 = \frac{8\pi R_\odot^2}{15M_\odot}r^4\rho,\ C_4 = \frac{64\pi^2 R_\odot^4}{9M_\odot^2}\frac{r^6\rho^2 D_\mathrm{v}}{\sqrt{\nu}}$$

7.9.7 Expressions initiales de $\breve{E}_\Omega$ et $\breve{E}_\mu$

Au paragraphe suivant, dans les expressions initiales de Maeder & Zahn [1998], pour E_Ω et E_μ des termes ont été regroupés. Pour référence, on reproduit les équations originales notées $\breve{E}_\Omega$ pour $\tilde{E}_\Omega$ et $\breve{E}_\mu$ pour $\tilde{E}_\mu$; $\tilde{E}_\Omega$ et $\tilde{E}_\mu$ sont définis plus avant.

$$\begin{aligned}
\breve{E}_\Omega = \quad & -\frac{\rho_m}{\rho}\left\{\frac{R}{3}\frac{\partial}{\partial R}\left[H_\mathrm{T}\frac{\partial}{\partial R}\left(\frac{\theta}{\delta}\right) + \left(1 - \frac{\chi_\mathrm{T}+1}{\delta}\right)\theta\right] - \frac{2H_\mathrm{T}}{R}\left(1 + \frac{D_\mathrm{h}}{K}\right)\frac{\theta}{\delta} + \frac{2}{3}\theta\right\} \\
& - \frac{\bar{\varepsilon} + \bar{\varepsilon}_g}{\varepsilon_\mathrm{m}}\left\{H_\mathrm{T}\frac{\partial}{\partial R}\left(\frac{\theta}{\delta}\right) + \left[f_\varepsilon\left(\frac{\varepsilon_T}{\delta} - 1\right) + 2 - \frac{\chi_\mathrm{T}+1}{\delta}\right]\theta\right\}
\end{aligned}$$

$$\begin{aligned}
\breve{E}_\mu = \quad & \frac{\rho_m}{\rho}\left\{\frac{R}{3}\frac{\partial}{\partial R}\left[H_\mathrm{T}\frac{\partial}{\partial R}\left(\frac{\varphi}{\delta}\Lambda\right) - \left(\chi_\mu + \frac{\chi_\mathrm{T}+1}{\delta}\varphi\right)\Lambda\right] - \frac{2H_\mathrm{T}}{R}\frac{\varphi}{\delta}\Lambda\right\} \\
& + \frac{\bar{\varepsilon} + \bar{\varepsilon}_g}{\varepsilon_\mathrm{m}}\left[H_\mathrm{T}\frac{\partial}{\partial R}\left(\frac{\varphi}{\delta}\Lambda\right) + \left(f_\varepsilon\varepsilon_\mu + f_\varepsilon\varepsilon_T\frac{\varphi}{\delta} - \chi_\mu - \frac{\chi_\mathrm{T}+1}{\delta}\varphi\right)\Lambda\right]
\end{aligned}$$

7.9.8 Expression vérifiée par E_Ω

Les quantités concernées[20] de $\breve{E}_\Omega$ et $\breve{E}_\mu$ ont été regroupées dans E_Ω.

$$
\begin{aligned}
E_\Omega = \quad & 2\left(1-\frac{\Omega^2}{2\pi G\rho}+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\bar{\varepsilon}+\bar{\varepsilon}_g}{\varepsilon_\mathrm{m}}\right)\frac{\tilde{g}}{g}+\tilde{E_\Omega}\\
= \quad & \frac{8R_\odot^3}{3GM_\odot}\left(1-\frac{\Omega^2}{2\pi G\rho}+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\partial\ln l}{\partial\ln m}\right)\frac{r^3\Omega^2}{\nu^{\frac{3}{2}}}+\tilde{E_\Omega}\\
\tilde{E_\Omega} = \quad & \frac{\rho_m}{\rho}\left\{\frac{R}{3}\frac{\partial}{\partial R}\left[H_\mathrm{T}\frac{\partial}{\partial R}\left(\frac{\varphi\Lambda-\theta}{\delta}\right)\right.\right.\\
& \left.-\left(1+\chi_\mathrm{T}\right)\frac{\varphi\Lambda-\theta}{\delta}-\theta-\chi_\mu\Lambda\right]\\
& \left.-2\frac{H_\mathrm{T}}{R}\left(\frac{\varphi\Lambda-\theta}{\delta}+\frac{R\theta}{3H_\mathrm{T}}-\frac{D_\mathrm{h}}{K}\frac{\theta}{\delta}\right)\right\}\\
\tilde{E_\Omega} = \quad & \frac{3M_\odot}{4\pi R_\odot^3}\frac{\nu^{\frac{3}{2}}}{r^3\rho}\left\{\frac{8\pi R_\odot^3}{9M_\odot}\frac{r^3\rho}{\sqrt{\nu}}\frac{\partial}{\partial\nu}\left[\frac{8\pi R_\odot^2}{3M_\odot}\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\frac{\partial}{\partial\nu}\psi\right.\right.\\
& \left.-\left(1+\chi_\mathrm{T}\right)\psi-\theta-\chi_\mu\Lambda\right]\\
& \left.-2\frac{H_\mathrm{T}}{R_\odot r}\left(\psi+\frac{R_\odot}{3}\frac{r\theta}{H_\mathrm{T}}-\frac{D_\mathrm{h}}{K}\frac{\theta}{\delta}\right)\right\}\\
\tilde{E_\Omega} = \quad & \nu\frac{16\pi R_\odot^2}{9M_\odot}\frac{\partial}{\partial\nu}\left(\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}\right)\\
& +\frac{2}{3}\nu\frac{\partial}{\partial\nu}\left\{-\left(1+\chi_\mathrm{T}\right)\psi-\theta-\chi_\mu\Lambda\right\}\\
& -\nu\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_\mathrm{T}\sqrt{\nu}}{r^4\rho}\left[\psi+\left(\frac{R_\odot r}{3H_\mathrm{T}}-\frac{D_\mathrm{h}}{K\delta}\right)\theta\right]
\end{aligned}
$$

7.9.9 Expression vérifiée par E_μ

Les quantités concernées de $\breve{E}_\Omega$ et $\breve{E}_\mu$ ont été regroupées dans E_μ.

$$
\begin{aligned}
E_\mu = \quad & \frac{\bar{\varepsilon}+\bar{\varepsilon}_g}{\varepsilon_\mathrm{m}}\left[H_\mathrm{T}\frac{\partial}{\partial R}\left(\frac{\varphi\Lambda-\theta}{\delta}\right)\right.\\
& \left.+\left(f_\varepsilon\varepsilon_T-\chi_\mathrm{T}-1\right)\frac{\varphi\Lambda-\theta}{\delta}-\left(2-f_\varepsilon\right)\theta+\left(f_\varepsilon\varepsilon_\mu-\chi_\mu\right)\Lambda\right]=\\
& \nu\frac{8\pi R_\odot^2}{3M_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^2\rho H_\mathrm{T}}{\nu^{\frac{3}{2}}}\frac{\partial\psi}{\partial\nu}+\nu\frac{M_\odot}{L_\odot}\frac{\varepsilon_{nuc}}{l}\sqrt{\nu}\left(\varepsilon_T\psi+\theta+\varepsilon_\mu\Lambda\right)\\
& -\nu\frac{\partial\ln l}{\partial\ln m}\frac{1}{\nu}\left[\left(\chi_\mathrm{T}+1\right)\psi+2\theta+\chi_\mu\Lambda\right]
\end{aligned}
$$

[20]La quantité $\frac{\Omega_0^2}{2\pi G\rho_0}$ a été introduite par S.Mathis & JP. Zahn, pour éliminer une singularité sur U au centre, quand le cœur est radiatif.

7.9.10 Expression vérifiée par $U = y_2$

Dans les zones radiatives :

$$\frac{M}{L}\left[\frac{Uc_pT}{H_p}\left(\nabla_{ad}-\nabla+\frac{\varphi}{\delta}\nabla_\mu\right)-\frac{c_pT}{\delta}\frac{D\theta}{Dt}\right] \tag{7.72}$$

$$-2\left(1-\frac{\Omega^2}{2\pi G\rho}+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\bar{\varepsilon}+\bar{\varepsilon}_g}{\varepsilon_{\mathrm{m}}}\right)\frac{\tilde{g}}{g}=\tilde{E_\Omega}+E_\mu\Rightarrow \tag{7.73}$$

$$\frac{M_\odot}{L_\odot}\frac{\nu^{\frac{3}{2}}}{l}\left[\frac{c_pT}{H_p}\left(\nabla_{ad}-\nabla+\frac{\varphi}{\delta}\nabla_\mu\right)U-\frac{c_pT}{\delta}\frac{\theta-\theta(t)}{\Delta t}\right]$$
$$-\frac{8R_\odot^3}{3GM_\odot}\left(1-\frac{\Omega^2}{2\pi G\rho}+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\partial\ln l}{\partial\ln m}\right)\frac{r^3\Omega^2}{\nu^{\frac{3}{2}}}=$$
$$\nu\left\{\frac{16\pi R_\odot^2}{9M_\odot}\frac{\partial}{\partial\nu}\left(\frac{r^2\rho H_{\mathrm{T}}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}\right)-\frac{2}{3}\frac{\partial}{\partial\nu}\left[(1+\chi_{\mathrm{T}})\,\psi+\theta+\chi_\mu\Lambda\right]\right.$$
$$+\frac{8\pi R_\odot^2}{3M_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^2\rho H_{\mathrm{T}}}{\nu^{\frac{3}{2}}}\frac{\partial\psi}{\partial\nu}-\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_{\mathrm{T}}\sqrt{\nu}}{r^4\rho}\left[\psi+\left(\frac{R_\odot r}{3H_{\mathrm{T}}}-\frac{D_{\mathrm{h}}}{K\delta}\right)\theta\right]$$
$$\left.+\frac{M_\odot}{L_\odot}\frac{\varepsilon_{nuc}}{l}\sqrt{\nu}\left(\varepsilon_T\psi+\theta+\varepsilon_\mu\Lambda\right)-\frac{\partial\ln l}{\partial\ln m}\frac{1}{\nu}\left[(\chi_{\mathrm{T}}+1)\,\psi+2\theta+\chi_\mu\Lambda\right]\right\}$$

Les solutions faibles sont données par :

$$<\left\{\frac{M_\odot}{L_\odot}\frac{\nu^{\frac{3}{2}}}{l\nu}\left[\frac{c_pT}{H_p}\left(\nabla_{ad}-\nabla+\frac{\varphi}{\delta}\nabla_\mu\right)U-\frac{c_pT}{\delta}\frac{\theta-\theta(t)}{\Delta t}\right]\right.$$
$$-\frac{8R_\odot^3}{3GM_\odot}\left(1-\frac{\Omega^2}{2\pi G\rho}+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\partial\ln l}{\partial\ln m}\right)\frac{r^3\Omega^2}{\nu^{\frac{5}{2}}}$$
$$-\frac{8\pi R_\odot^2}{3M_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^2\rho H_{\mathrm{T}}}{\nu^{\frac{3}{2}}}\frac{\partial\psi}{\partial\nu}+\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_{\mathrm{T}}\sqrt{\nu}}{r^4\rho}\left[\psi+\left(\frac{R_\odot r}{3H_{\mathrm{T}}}-\frac{D_{\mathrm{h}}}{K\delta}\right)\theta\right]$$
$$\left.-\frac{M_\odot}{L_\odot}\frac{\varepsilon_{nuc}}{l}\sqrt{\nu}\left(\varepsilon_T\psi+\theta+\varepsilon_\mu\Lambda\right)+\frac{\partial\ln l}{\partial\ln m}\frac{1}{\nu}\left[(\chi_{\mathrm{T}}+1)\,\psi+2\theta+\chi_\mu\Lambda\right]\right\}\bullet\ S_j^{m_c}>$$
$$+<\left\{\frac{16\pi R_\odot^2}{9M_\odot}\frac{r^2\rho H_{\mathrm{T}}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}-\frac{2}{3}\left[(1+\chi_{\mathrm{T}})\,\psi+\theta+\chi_\mu\Lambda\right]\right\}\bullet\ \frac{\partial S_j^{m_c}}{\partial\nu}>$$
$$-\left|\left\{\frac{16\pi R_\odot^2}{3M_\odot}\frac{r^2\rho H_{\mathrm{T}}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}-\frac{2}{3}\left[(1+\chi_{\mathrm{T}})\,\psi+\theta+\chi_\mu\Lambda\right]\right\}S_j^{m_c}\right|_{\nu_{\mathrm{c}}}^{\nu_{\mathrm{s}}}=0$$
$$\Longrightarrow$$
$$<\left\{\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{lH_p}\left(\nabla_{ad}-\nabla+\frac{\varphi}{\delta}\nabla_\mu\right)U-\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{l\delta}\frac{\theta-\theta(t)}{\Delta t}\right.$$
$$-\frac{8R_\odot^3}{3GM_\odot}\frac{r^3}{\nu^{\frac{5}{2}}}\Omega^2+\frac{4R_\odot^3}{3\pi G^2M_\odot}\frac{r^3}{\rho\nu^{\frac{5}{2}}}\Omega^4-\frac{4R_\odot^3\Omega_0^2}{3\pi G^2M_\odot\rho_0}\frac{r^3}{\nu^{\frac{5}{2}}}\Omega^2+\frac{8R_\odot^3}{3GM_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^3}{\nu^{\frac{5}{2}}}\Omega^2$$
$$+\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_{\mathrm{T}}\sqrt{\nu}}{r^4\rho}\psi-\frac{8\pi R_\odot^2}{3M_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^2\rho H_{\mathrm{T}}}{\nu^{\frac{3}{2}}}\frac{\partial\psi}{\partial\nu}+\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_{\mathrm{T}}\sqrt{\nu}}{r^4\rho}\left(\frac{R_\odot}{3}\frac{r}{H_{\mathrm{T}}}-\frac{D_{\mathrm{h}}}{K\delta}\right)\theta$$
$$-\frac{M_\odot}{L_\odot}\frac{\varepsilon_{\mathrm{nuc}}\varepsilon_{\mathrm{T}}\sqrt{\nu}}{l}\psi-\frac{M_\odot}{L_\odot}\frac{\varepsilon_{\mathrm{nuc}}\sqrt{\nu}}{l}\theta-\frac{M_\odot}{L_\odot}\frac{\varepsilon_{\mathrm{nuc}}\varepsilon_\mu\sqrt{\nu}}{l}\Lambda+\frac{\partial\ln l}{\partial\ln m}\frac{\chi_{\mathrm{T}}+1}{\nu}\psi$$
$$\left.+\frac{2}{\nu}\frac{\partial\ln l}{\partial\ln m}\theta+\frac{\partial\ln l}{\partial\ln m}\frac{\chi_\mu}{\nu}\Lambda\right\}\bullet\ S_j^{m_c}>$$

$$
\begin{aligned}
&+<\left\{\frac{16\pi R_\odot^2}{9M_\odot}\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}-\frac{2}{3}\left[(1+\chi_\mathrm{T})\,\psi+\theta+\chi_\mu\Lambda\right]\right\}\bullet\ \frac{\partial S_j^{m_c}}{\partial\nu}>\\
&-\left|\left\{\frac{16\pi R_\odot^2}{9M_\odot}\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}-\frac{2}{3}\left[(1+\chi_\mathrm{T})\,\psi+\theta+\chi_\mu\Lambda\right]\right\}S_j^{m_c}\right|_{\nu_\mathrm{c}}^{\nu_\mathrm{s}}=0\\
&\Longrightarrow\\
&<\left\{\frac{4R_\odot^3}{3\pi G^2M_\odot}\frac{r^3}{\rho\nu^{\frac{5}{2}}}\Omega^4-\frac{8R_\odot^3}{3GM_\odot}\frac{r^3}{\nu^{\frac{5}{2}}}\left(1+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\partial\ln l}{\partial\ln m}\right)\Omega^2\right.\\
&+\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{lH_p}\left(\nabla_{ad}-\nabla+\frac{\varphi}{\delta}\nabla_\mu\right)U-\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{l\delta\Delta t}\theta+\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}\theta(t)}{l\delta\Delta t}\\
&+\left(\frac{M_\odot}{2\pi R_\odot^3}\frac{\sqrt{\nu}}{r^3\rho}-\frac{M_\odot}{L_\odot}\frac{\varepsilon_\mathrm{nuc}\sqrt{\nu}}{l}+\frac{2}{\nu}\frac{\partial\ln l}{\partial\ln m}-\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_\mathrm{T}\sqrt{\nu}}{r^4\rho K\delta}D_\mathrm{h}\right)\theta\\
&+\left(\frac{\partial\ln l}{\partial\ln m}\frac{\chi_\mu}{\nu}-\frac{M_\odot}{L_\odot}\frac{\varepsilon_\mathrm{nuc}\varepsilon_\mu\sqrt{\nu}}{l}\right)\Lambda+\left(\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_\mathrm{T}\sqrt{\nu}}{r^4\rho}-\frac{M_\odot}{L_\odot}\frac{\varepsilon_\mathrm{nuc}\varepsilon_\mathrm{T}\sqrt{\nu}}{l}+\frac{\partial\ln l}{\partial\ln m}\frac{\chi_\mathrm{T}+1}{\nu}\right)\psi\\
&\left.-\frac{8\pi R_\odot^2}{3M_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^2\rho H_\mathrm{T}}{\nu^{\frac{3}{2}}}\frac{\partial\psi}{\partial\nu}\right\}\bullet\ S_j^{m_c}>\\
&+<\left\{\frac{16\pi R_\odot^2}{9M_\odot}\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}-\frac{2}{3}\left[(1+\chi_\mathrm{T})\,\psi+\theta+\chi_\mu\Lambda\right]\right\}\bullet\ \frac{\partial S_j^{m_c}}{\partial\nu}>\\
&-\left|\left\{\frac{16\pi R_\odot^2}{9M_\odot}\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\frac{\partial\psi}{\partial\nu}-\frac{2}{3}\left[(1+\chi_\mathrm{T})\,\psi+\theta+\chi_\mu\Lambda\right]\right\}S_j^{m_c}\right|_{\nu_\mathrm{c}}^{\nu_\mathrm{s}}=0
\end{aligned}
$$

Avec les notations des algorithmes, les équations à résoudre s'écrivent :

$$
\begin{aligned}
&<\left\{\left(C_5\Omega^2-C_7\right)\Omega^2+C_{10}+C_6U+C_{11}\theta+C_{15}\Lambda+C_{12}\psi-C_{13}\frac{\partial\psi}{\partial\nu}\right\}\bullet\ S_j^{m_c}>\\
&+<\left(C_{17}\theta+C_{18}\Lambda+C_{16}\psi+C_{14}\frac{\partial\psi}{\partial\nu}\right)\bullet\ \frac{\partial S_j^{m_c}}{\partial\nu}>\\
&-\left|\left(C_{17}\theta+C_{18}\Lambda+C_{16}\psi+C_{14}\frac{\partial\psi}{\partial\nu}\right)S_j^{m_c}\right|_{\nu_\mathrm{c}}^{\nu_\mathrm{s}}=0
\end{aligned}
\tag{7.74}
$$

$$
\begin{aligned}
&C_5=\frac{4R_\odot^3}{3\pi G^2M_\odot}\frac{r^3}{\rho\nu^{\frac{5}{2}}},\quad C_6=\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{lH_p}\left|\nabla_{ad}-\nabla+\frac{\varphi}{\delta}\nabla_\mu\right|,\\
&C_7=\frac{8R_\odot^3}{3GM_\odot}\frac{r^3}{\nu^{\frac{5}{2}}}\left(1+\frac{\Omega_0^2}{2\pi G\rho_0}-\frac{\partial\ln l}{\partial\ln m}\right),\quad C_{10}=\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{l\delta\Delta t}\theta(t)\\
&C_{11}=C_2^*-C_3^*+C_4^*-C_5^*-C_{16}^*,\quad C_2^*=\frac{M_\odot}{2\pi R_\odot^3}\frac{\sqrt{\nu}}{r^3\rho},\quad C_3^*=\frac{M_\odot}{L_\odot}\frac{\varepsilon_\mathrm{nuc}\sqrt{\nu}}{l}\\
&C_4^*=\frac{2}{\nu}\frac{\partial\ln l}{\partial\ln m},\quad C_5^*=\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_\mathrm{T}\sqrt{\nu}D_\mathrm{h}}{r^4\rho K\delta},\quad C_{16}^*=\frac{M_\odot}{L_\odot}\frac{c_pT\sqrt{\nu}}{l\delta\Delta t}\\
&C_{12}=C_6^*-C_7^*+C_8^*,\quad C_6^*=\frac{3M_\odot}{2\pi R_\odot^4}\frac{H_\mathrm{T}\sqrt{\nu}}{r^4\rho},\quad C_7^*=\frac{M_\odot}{L_\odot}\frac{\varepsilon_\mathrm{nuc}\varepsilon_\mathrm{T}\sqrt{\nu}}{l}\\
&C_8^*=\frac{\partial\ln l}{\partial\ln m}\frac{\chi_\mathrm{T}+1}{\nu},\quad C_{13}=\frac{8\pi R_\odot^2}{3M_\odot}\frac{\partial\ln l}{\partial\ln m}\frac{r^2\rho H_\mathrm{T}}{\nu^{\frac{3}{2}}},\quad C_{14}=\frac{16\pi R_\odot^2}{9M_\odot}\frac{r^2\rho H_\mathrm{T}}{\sqrt{\nu}}\\
&C_{15}=C_9^*-C_{10}^*,\quad C_9^*=\frac{\partial\ln l}{\partial\ln m}\frac{\chi_\mu}{\nu},\quad C_{10}^*=\frac{M_\odot}{L_\odot}\frac{\varepsilon_\mathrm{nuc}\varepsilon_\mu\sqrt{\nu}}{l}
\end{aligned}
$$

$$C_{16} = -\frac{2}{3}\left(1+\chi_{\rm T}\right), \quad C_{17} = -\frac{2}{3}, \quad C_{18} = -\frac{2}{3}\chi_\mu$$

7.9.11 Expression vérifiée par $\theta = y_3$

$$\theta = \frac{\tilde{\rho}}{\bar{\rho}} = \frac{R^2}{3g}\frac{\partial \Omega^2}{\partial R} \tag{7.75}$$

$$\theta = \frac{R_\odot^2 r^2 R_\odot^2 r^2 8\pi R_\odot^2 r^2 \rho 2\Omega}{3GM_\odot \nu^{\frac{3}{2}} 3M_\odot\sqrt{\nu}}\frac{\partial\Omega}{\partial\nu} = \frac{16\pi R_\odot^6}{9GM_\odot^2}\frac{r^6\rho\Omega}{\nu^2}\frac{\partial\Omega}{\partial\nu} \Rightarrow$$
$$< \left(\theta - \frac{16\pi R_\odot^6}{9GM_\odot^2}\frac{r^6\rho}{\nu^2}\Omega\frac{\partial\Omega}{\partial\nu}\right)\bullet S_j^{m_c} >= 0$$

Avec les notations des algorithmes, en tenant compte des conditions limites, $\frac{\partial\Omega}{\partial\nu} = 0$, au centre ($\nu = \nu_{\rm c}$) et à la surface ($\nu = \nu_{\rm s}$), les solutions faibles vérifient :

$$< \left(\theta - C_{19}\Omega\frac{\partial\Omega}{\partial\nu}\right)\bullet S_j^{m_c} >= 0, \quad C_{19} = \frac{16\pi R_\odot^6}{9GM_\odot^2}\frac{r^6\rho}{\nu^2}$$

7.9.12 Expression vérifiée par $\Lambda = y_4$

L'expression vérifiée par Λ est un problème de valeur initiale :

$$\frac{D\Lambda}{Dt} = \frac{\nabla_\mu U}{H_{\rm P}} - \frac{6D_{\rm h}\Lambda}{R^2} \tag{7.76}$$

Les solutions faibles sont données par :

$$< \left(\frac{\Lambda - \Lambda(t)}{\Delta t} - \frac{\nabla_\mu}{H_{\rm P}}U + \frac{6D_{\rm h}}{R_\odot^2 r^2}\Lambda\right)\bullet S_j^{m_c} >= 0$$

ou encore :

$$< \left(C_{20}\Lambda - C_{22} - C_{21}U\right)\bullet S_j^{m_c} >= 0$$
$$C_{20} = C_1^* + C_{13}^*, \quad C_1^* = \frac{1}{\Delta t}, \quad C_{13}^* = \frac{6D_{\rm h}}{R_\odot^2 r^2}, \quad C_{21} = \frac{\nabla_\mu}{H_{\rm P}}, \quad C_{22} = \frac{\Lambda(t)}{\Delta t}$$

7.9.13 Expression vérifiée par $\psi = y_5 = \psi$

$$\psi = \frac{\varphi\Lambda - \theta}{\delta} \tag{7.77}$$

Les solutions faibles sont données par :

$$< \left(\psi - C_{24}\Lambda + C_{23}\theta\right)\bullet S_j^{m_c} >= 0$$
$$C_{24} = \frac{\varphi}{\delta}, \quad C_{23} = \frac{1}{\delta}$$

7.9.14 Transport des abondances par mole x_i

Pour les n_c éléments chimiques d'abondance par mole *i.e.* de concentration $x_i,\ i=1,\dots,c$, l'équation de transport s'écrit, Matis & Zahn [2004], équation B2 :

$$\rho\left(\frac{Dx_i}{Dt}-\Upsilon_i\right)=\frac{1}{R^2}\frac{\partial}{\partial R}\left[R^2\rho\left(D_i\bullet\nabla_R\mathcal{X}-x_iv_i\right)\right],\ \nabla_R\mathcal{X}\equiv\left(\frac{\partial x_1}{\partial R},\dots,\frac{\partial x_{n_c}}{\partial R}\right)^{\mathrm{T}}$$
$$D_i\equiv\left(-x_ib_{i1},\dots,-x_ib_{ii}+D_{\mathrm{M}}+D_{\mathrm{eff}},\dots,-x_ib_{in_c}\right)^{\mathrm{T}}$$

les b_{ij} et v_i sont obtenus en résolvant l'équation de Burgers pour la diffusion microscopique. Les coefficients de diffusion turbulente D_{eff} et de mélange D_{M} affectent le terme diagonal b_{ii}. $\Upsilon_i=\left(\frac{Dx_i}{Dt}\right)_{\mathrm{nucl}}$ est le taux de variation temporelle de l'élément chimique i dû aux réactions thermonucléaires.

Avec la variable lagrangienne ν :

$$\frac{Dx_i}{Dt}-\Upsilon_i=\frac{8\pi R_\odot^2r^2\rho}{3M_\odot R_\odot^2r^2\rho\sqrt{\nu}}\frac{\partial}{\partial\nu}\left[\frac{8\pi R_\odot^4r^4\rho^2}{3M_\odot\sqrt{\nu}}D_i\bullet\nabla_\nu\mathcal{X}-R_\odot^2r^2\rho x_iv_i\right]\Rightarrow$$
$$\frac{3}{2}\sqrt{\nu}\left(\frac{Dx_i}{Dt}-\Upsilon_i\right)=\frac{32\pi^2R_\odot^4}{3M_\odot^2}\frac{\partial}{\partial\nu}\left(\frac{r^4\rho^2}{\sqrt{\nu}}D_i\bullet\nabla_\nu\mathcal{X}\right)-\frac{4\pi R_\odot^2}{M_\odot}\frac{\partial}{\partial\nu}\left(r^2\rho x_iv_i\right)$$

L'intégration par éléments finis prend la forme :

$$<\frac{3}{2}\sqrt{\nu}\left(x_i-x_i(t)-\Upsilon_i\Delta t\right)\bullet S_j^{m_c}>+$$
$$<\left[\frac{32\pi^2R_\odot^4}{3M_\odot^2}\frac{r^4\rho^2\Delta t}{\sqrt{\nu}}\left(D_i\bullet\nabla_\nu\mathcal{X}\right)-\frac{4\pi R_\odot^2}{M_\odot}r^2\rho\Delta t x_iv_i\right]\bullet\frac{\partial S_j^{m_c}}{\partial\nu}>-$$
$$\left|\left(\frac{32\pi^2R_\odot^4}{3M_\odot^2}\frac{r^4\rho^2\Delta t}{\sqrt{\nu}}\left(D_i\bullet\nabla_\nu\mathcal{X}\right)-\frac{4\pi R_\odot^2}{M_\odot}r^2\rho\Delta t x_iv_i\right)S_j^{m_c}\right|_{\nu_{\mathrm{c}}}^{\nu_{\mathrm{s}}}=0$$

Avec les notations des algorithmes, en tenant compte des conditions limites, $\frac{\partial x_i}{\partial\nu}=0,\ v_i=0$ au centre ($\nu=\nu_0$) et à la surface ($\nu=\nu_{\mathrm{s}}$) :

$$<C_1\left(x_i-x_i(t)-\Upsilon_i\Delta t\right)\bullet S_j^{m_c}>+$$
$$<\left[C_2\left(D_i\bullet\nabla_\nu\mathcal{X}\right)-C_3x_iv_i\right]\bullet\frac{\partial S_j^{m_c}}{\partial\nu}>=0 \tag{7.78}$$
$$C_1=\frac{3}{2}\sqrt{\nu},\ C_2=\frac{32\pi^2R_\odot^4}{3M_\odot^2}\frac{r^4\rho^2\Delta t}{\sqrt{\nu}},\ C_3=\frac{4\pi R_\odot^2}{M_\odot}r^2\rho\Delta t$$

7.9.15 Les conditions physiques dans les zones mélangées

Les zones mélangées, *i.e.* zones convectives éventuellement overshootées, apparaissent, disparaissent au cours de l'évolution. On admet que dans ces zones le mélange uniformise la vitesse angulaire, c'est la rotation solide. On suppose aussi que les fluctuation de densité, température et composition chimiques sont nulles le long d'une isobare, ainsi :

$$\frac{\partial\Omega}{\partial R}=0,\ U=0,\ \theta=0,\ \Lambda=0,\ \psi=0,\ \frac{\partial\psi}{\partial R}. \tag{7.79}$$

Ces conditions entraînent la discontinuité des gradients des fonctions inconnues aux limites entre zones radiatives et zones mélangées. D'où la nécessité d'introduire des limites internes mobiles et en nombre variable, ce qui complique l'intégration numérique. Grâce à un choix *ad hoc* des bases de B-splines utilisées pour la diffusion du moment cinétique, ces discontinuités sont implicitement prises en compte par CESAM2k, *cf.* § 7.7.3 (Page 95). Une alternative se présente :

- ***Le modèle est totalement convectif***, à l'image de ce qui est fait pour la diffusion des espèces chimiques, on contourne l'aspect intégral en uniformisant la vitesse angulaire par une diffusion permettant de conserver globalement le moment cinétique. Pour ce faire, on impose des coefficients de diffusion dont l'ordre de grandeur, grand devant l'unité, correspond au temps de retournement des éléments convectifs. Cette modélisation simule un mélange turbulent efficace. Les équations à résoudre sont alors les suivantes :

$$< (C_1\Omega - C_2) \bullet S_j^{m_c} > + < C_4\frac{\partial\Omega}{\partial\nu} \bullet \frac{\partial S_j^{m_c}}{\partial\nu} > - \left| C_4\frac{\partial\Omega}{\partial\nu} S_j^{m_c} \right|_{\nu_c}^{\nu_s} = 0$$
$$< U \bullet S_j^{m_c} >= 0,\ < \theta \bullet S_j^{m_c} >= 0,\ < \Lambda \bullet S_j^{m_c} >= 0,\ < \Psi \bullet S_j^{m_c} >= 0.$$

  Les deux conditions limites de cette équation du second ordre, à savoir $\frac{\partial\Omega}{\partial\nu} = 0$ au centre et à la surface, s'introduisent naturellement dans la partie intégrée qui n'apporte aucune contribution. C'est la situation la plus simple, aucune discontinuité ne devant être introduite dans le vecteur nodal.
- ***Au moins une zone radiative est présente***, comme précédemment, dans la ou les zones convectives, l'uniformisation de la vitesse angulaire est effectuée par une diffusion. Contrairement au cas totalement convectif, le moment cinétique n'est pas globalement conservé en tant que tel. Sa variation temporelle dans chaque zone convective est supposée advectée dans la circulation méridienne des zones radiatives adjacentes.
  A la limite R d'une zone mélangée localisée dans un intervalle $[R_1, R]$, la vitesse de circulation méridienne vérifie (Mathis & Zahn [2004]) :

$$\int_{R_1}^{R} \left(\frac{D}{Dt} R^4\rho\Omega + R^2\rho\dot{\mathcal{M}}_\Omega \right) dR = \left| \frac{1}{5} R^4\rho\Omega U \right|_{R_1}^{R}, \tag{7.80}$$

  différence entre les quantités d'énergie entrant et sortant de la zone mélangée par unité de temps. On a donc :

$$\frac{D}{Dt} R^4\rho\Omega + R^2\rho\dot{\mathcal{M}}_\Omega = \frac{1}{5}\frac{\partial}{\partial R} R^4\rho\Omega U \tag{7.81}$$

  La résolution de cette équation conduit à la connaissance de la valeur de la vitesse de circulation méridienne coté radiatif des limites zone radiative / zone convective. Comme le montre l'équation 7.80 (Page 121), U doit être affecté du signe + à la "sortie" d'une zone convective et du signe − à son "entrée" (Mathis & Zahn [2004]).
  Du point de vue algorithmique, la vitesse de circulation méridienne n'intervenant pas dans le processus de mélange par diffusion, pour résoudre l'équation 7.81 (Page 121), CESAM2k n'introduit pas de variable supplémentaire mais utilise celle attachée à U. Pour la suite, cette variable,

dénuée de sens physique, sera notée $U_{\rm f}$ pour la distinguer de la vitesse de circulation méridienne U, nulle dans une zone convective.
Les équations sont discrétisées comme suit :

$$\frac{D}{Dt}R^4\rho\Omega + R^2\rho\dot{\mathcal{M}}_\Omega = \frac{1}{5}\frac{\partial}{\partial R}R^4\rho\Omega U \Rightarrow R_\odot^4\frac{D}{Dt}r^4\rho\Omega + R_\odot^2 r^2\rho\dot{\mathcal{M}}_\Omega = \frac{8\pi R_\odot^6}{15M_\odot}\frac{r^2\rho}{\sqrt{\nu}} \Rightarrow$$
$$\sqrt{\nu}\frac{D}{Dt}r^2\Omega + \frac{\sqrt{\nu}\Omega}{\rho}\frac{D}{Dt}r^2\rho + \sqrt{\nu}\frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2} = \frac{8\pi R_\odot^6}{15M_\odot}\frac{r^2\rho}{\sqrt{\nu}} \Rightarrow$$
$$\sqrt{\nu}\left(\frac{r^2\Omega}{\Delta t} - \frac{r^2(t)\Omega(t)}{\Delta t} + \frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2}\right) + \frac{\sqrt{\nu}\Omega}{\rho}\left(\rho\frac{D}{Dt}r^2 + r^2\frac{D}{Dt}\rho\right) = \frac{8\pi R_\odot^6}{15M_\odot}\frac{r^2\rho}{\sqrt{\nu}}$$

En tenant compte du fait que le gradient de composition chimique est nul dans une zone convective :

$$\frac{D\ln\rho}{Dt} = \alpha\frac{D\ln P}{Dt} - \delta\frac{D\ln T}{Dt} + \varphi\frac{D\ln\mu}{Dt} = (\alpha - \delta\nabla)\frac{D\ln P}{Dt} \Rightarrow$$
$$\frac{\sqrt{\nu}}{\Delta t}\left[r^2\left\{2 + (\alpha - \delta\nabla)(\ln P - \ln P(t))\right\} - r^2(t)\right]\Omega$$
$$-\sqrt{\nu}\left(\frac{r^2(t)\Omega(t)}{\Delta t} - \frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2}\right) = \frac{8\pi R_\odot^6}{15M_\odot}\frac{r^2\rho}{\sqrt{\nu}}$$

Les équations 7.75 (Page 119) et 7.76 (Page 119) montrent que θ et Λ sont discontinues aux limites zone radiative / zone convective. Ces discontinuités entraînent celle de ψ, *cf.* équation 7.77 (Page 119). En résumé, aux limites zone radiative / zone convective :
– Ω est une fonction continue non dérivable,
– U, prolongeant $U_{\rm f}$, devient une fonction continue,
– θ, Λ et Ψ sont des fonctions discontinues.
Pour la diffusion du moment cinétique CESAM2k utilise un vecteur nodal discontinu à chaque limite zone radiative / zone convective ; la continuité de Ω et de U étant réalisée en assurant l'égalité de leurs valeurs de part et d'autre de chaque discontinuité, *cf.* § 7.7.3 (Page 95).
Avec les notations des algorithmes dans une zone mélangée d'un modèle non totalement convectif, les équations s'écrivent :

$$< (C_1\Omega - C_2)\bullet S_j^{m_c} > + < C_4\frac{\partial\Omega}{\partial\nu}\bullet\frac{\partial S_j^{m_c}}{\partial\nu} > - \left|C_4\frac{\partial\Omega}{\partial\nu}S_j^{m_c}\right|_{\nu_{\rm c}}^{\nu_{\rm s}} = 0$$
$$< (C_8\Omega - C_2)\bullet S_j^{m_c} > + < C_3\Omega U_{\rm f}\bullet\frac{\partial S_j^{m_c}}{\partial\nu} >$$
$$- \left|C_3\Omega U_{\rm f}S_j^{m_c}\right|_{\nu_{\rm c}}^{\nu_{\rm s}} = 0 \qquad (7.82)$$
$$< \theta\bullet S_j^{m_c} >= 0,\ < \Lambda\bullet S_j^{m_c} >= 0,\ < \Psi\bullet S_j^{m_c} >= 0$$
$$C_1 = \frac{r^2\sqrt{\nu}}{\Delta t},\ C_2 = (C_{11}^* - C_{12}^*)\sqrt{\nu},\ C_{11}^* = \frac{r^2(t)\Omega(t)}{\Delta t},\ C_{12}^* = \frac{\dot{\mathcal{M}}_\Omega}{R_\odot^2}$$
$$C_4 = \frac{64\pi^2R_\odot^4}{9M_\odot^2}\frac{r^6\rho^2D_{\rm v}}{\sqrt{\nu}},\ C_8 = \frac{\sqrt{\nu}}{\Delta t}\left(r^2\left(2 + (\alpha - \delta\nabla)(\ln P - \ln P(t)) - r^2(t)\right)\right).$$

Les bornes des parties intégrées correspondent aux limites de la zone convective, *cf.* § 7.7.3 (Page 95).

7.9.16 Les conditions physiques aux limites

En taitant la nature intégrale du mélange convectif comme une diffusion, dans son ensemble, le transport du moment cinétique est ramené à un problème différentiel aux limites de conditions initiales. Des conditions initiales et des conditions limites doivent être précisées à chaque limite entre zone radiative et zone convective.
La diffusion du moment cinétique, *cf.* équation 7.70 (Page 114), est un problème aux limites du second ordre. La connaissance du profil initial de la vitesse angulaire et deux conditions limites spatiales sont nécessaires.
La vitesse de circulation méridienne, *cf.* équation 7.72 (Page 117), contitue un problème aux limites du quatrième ordre. La connaissance du profil initial de la vitesse de circulation méridienne et quatre conditions limites spatiales sont requises.
Le taux de fluctuation du poids moléculaire moyen le long d'une isobare, Λ, vérifie un problème de condition initiale, *cf.* équation 7.76 (Page 119). Seule la dépendance spatiale initiale de Λ doit être connue.
Les conditions initiales et aux limites des variables intermédiaires θ et ψ, se déduisent de celles de Ω, U et Λ, *cf.* équation 7.75 (Page 119) et 7.77 (Page 119).
Les conditions initiales et aux limites du système différentiel modélisant le transport des éléments chimiques ont été décrites au § 7.8 (Page 96).

Le centre : ($\nu = \nu_0 = 0$), suivant l'état d'évolution le centre peut être situé dans une zone radiative ou dans une zone convective. Par raison de symétrie, la dérivée spatiale de la vitesse angulaire est nulle, ainsi que la vitesse de circulation méridienne :

$$\frac{\partial\Omega(\nu_0,t)}{\partial\nu} = 0,\ U(\nu_0,t) = 0,$$

ces conditions entraînent, *cf.* équation 7.75 (Page 119) :

$$\theta(\nu_0,t) = 0.$$

Le rayon nul et l'échelle de hauteur de la pression infinie au centre entraînent, *cf.* équation 7.76 (Page 119) :

$$\Lambda(\nu_0,t) = 0.$$

Des conditions précédentes résulte :

$$\psi(\nu_0,t) = 0,\ \frac{\partial\psi(\nu_0,t)}{\partial\nu} = 0.$$

En présence d'un noyau convectif, on a aussi :

$$U_{\rm f}(\nu_0,t) = 0.$$

Avec ces conditions limites, les contributions des parties intégrées des équations 7.71 (Page 115), 7.82 (Page 122) et 7.74 (Page 118) sont nulles en ν_0.

La surface : La partie externe de l'enveloppe, zone d'ionisation de l'hydrogène, est toujours une zone convective. L'instabilité convective s'étend dans l'atmosphère jusqu'à la limite du milieu transparent ($\tau \sim \frac{2}{3}$). Il est raisonable d'admettre que la composition chimique et la vitesse angulaire dans l'atmosphère sont celles de la zone convective externe, *cf.* § 7.3.2 (Page 76). La borne supérieure $\nu_{\rm s}$ de l'intégration

spatiale du transport du moment cinétique correspond à celle de l'enveloppe.
A la surface les conditions limites retenues sont :

$$\frac{\partial\Omega(\nu_{\rm s},t)}{\partial\nu}=0,\ U(\nu_{\rm s},t)=0,\ \frac{\partial\theta(\nu_{\rm s},t)}{\partial\nu}=0,\ \frac{\partial\Lambda(\nu_{\rm s},t)}{\partial\nu}=0,\ \frac{\partial\psi(\nu_{\rm s},t)}{\partial\nu}=0.$$

Avec ces conditions limites, les contributions des parties intégrées des équations 7.71 (Page 115), 7.82 (Page 122) et 7.74 (Page 118) sont nulles en $\nu_{\rm s}$, sauf celle qui concerne $U_{\rm f}$ car rien n'impose que $U_{\rm f}(\nu_{\rm s},t)=0$.

Les limites zone radiative / zone convective internes : les discontinuités étant explicitement introduites, les conditions limites doivent être prises en compte. Les deux conditions limites requises par l'équation de transport du moment cinétique, sont satisfaites en écrivant la continuité de la vitesse angulaire à travers les limites zones radiatives / zones convectives. Pour ce faire, CESAM$2k$ remplace la relation résultant du produit scalaire avec la B-spline localisée sur la face convective de chaque limite, par la relation formée par l'égalisation des projections sur chaque B-spline située immédiatement de part et d'autre de la limite, *cf.* § 7.7.3 (Page 95). Pour chaque zone radiative interne deux conditions sont ainsi obtenues. Quand la zone radiative contient le centre, c'est la condition limite sur Ω qui est relevante.
De façon similaire deux conditions limites sont obtenues en prolongeant $U_{\rm f}$ par $\pm U$ sur chaque limite, *cf.* § 7.9.15 (Page 120).
Les contributions des parties intégrées de l'équation 7.74 (Page 118), complètent la quantité de conditions limites requises pour U.

7.9.17 Les conditions initiales

Des conditions initiales doivent être définies pour la vitesse angulaire Ω_0, la vitesse de circulation méridienne U_0 et le taux de fluctuation du poids moléculaire moyen Λ_0. Les conditions initiales sont établies par la routine PRIVATE `initialise_rota` du module `MOD_CESAM`.

Vitesse angulaire initiale : on ne dispose pas de données observationnelles permettant d'inférer la répartition spatiale de la vitesse angulaire d'un modèle initial de pré-séquence principale ou de séquence principale d'âge zéro. Faute de mieux, CESAM$2k$ suppose la rotation solide. La vitesse angulaire initiale est déduite de la variable `W_ROT` du fichier de données `mon_modele.don`, *cf.* § 3.2 (Page 21).
Il est toutefois possible d'utiliser un profil de rotation grossièrement similaire à celui que l'on peut inférer des observations de sismologie solaire :

$$\Omega_0(r,0)=\Omega_0\left\{1+q\left[1+\cos\left(\pi\frac{r}{R_\star}\right)\right]\right\}$$

Ω_0 est la vitesse angulaire initiale, éventuellement déduite de la période de rotation et du rayon total initial, *cf.* § 3.2 (Page 21). $q\geq -1$ est un paramètre permettant de fixer le rapport entre la vitesse angulaire au centre et à la surface, $q=0$ correspond à une rotation rigide. q est la variable `w_form` du module `MOD_DONNEES` elle est fixée à 0. Pour utiliser une valeur différente, il convient de la redéfinir à l'aide d'un fichier de "réglages", *cf.* § 4.8 (Page 43). Avec $q\neq 0$, le profil de Ω_0 n'est pas iniformisé dans les zone convective. Ω est initialisé dans la routine `w_initial` du module `MOD_CESAM`.

Vitesse de circulation méridienne initiale : cette quantité n'est pas accéssible à l'observation. De façon euristique, dans la routine `u_initial` du module `MOD_CESAM`, U_0 est obtenu par la relation empirique :

$$U_0(R) = \frac{|w_{\rm rot}|u_i}{2 - (R/R_\star)^2}$$

où u_i est un paramètre fixé à 10^{-4} et $w_{\rm rot}$ la valeur de la variable `W_ROT` du fichier de données `mon_modele.don`.

Taux de fluctuation du poids moléculaire moyen : Λ est supposé nul à l'instant initial : $\Lambda_0 = 0$.

Conditions initiales pour θ, et ψ : elles sont nécessaires pour la résolution numérique, la formulation des équations étant implicite. CESAM2k les calcule à partir des relations les définissant, *cf.* équation 7.75 (Page 119) et équation 7.77 (Page 119), en utilisant les valeurs initiales retenues pour Ω et Λ.

7.9.18 Pertes / gains de moment cinétique

Par l'intermédiaire de la routine générique `pertw` CESAM2k offre la possibilité d'utiliser différentes formes de pertes/gains de moment cinétique. Chacune d'elles dépend d'un paramètre libre, respectivement a, γ, ϖ, transmis par la varible `p_pertw` du fichier de données, *cf.* § 3.2 (Page 21). Une valeur négative/positive de ce paramètre correspond à une perte/gain de moment cinétique. Cette perte/gain de moment cinétique ne concerne que la zone convective externe. Les formes suivantes sont implantées dans CESAM2k :

- `pertw_sch` : La variation temporelle de moment cinétique par unité de masse et de temps, proportionnelle à Ω^3 (Schumanish), est modélisée par :

  $$\dot{\mathcal{M}}_\Omega = aR^2\Omega_{\rm s}^3, \quad |a| \sim 1.10^{-9}{\rm s}.$$

- `pertw_loc` : Perte / gain de moment cinétique proportionnel à l'énergie cinétique de rotation locale. La variation temporelle de moment cinétique par unité de masse est modélisée par :

  $$\dot{\mathcal{M}}_\Omega = \gamma R^2\Omega^2, \quad |\gamma| \sim 10^{-13} \sim 10^{-14}.$$

- `pertw_ptm` : Perte / gain de moment cinétique conséquence d'une perte / gain de masse $\dot{\mathcal{M}}$. Dans le cas d'une perte de masse, $\dot{\mathcal{M}} < 0$, on suppose qu'une couche externe se détache et emporte avec elle son moment cinétique, alors $\varpi = 1$. La perte de moment cinétique par unité de masse est approchée par :

  $$\dot{\mathcal{M}}_\Omega = \varpi\frac{\dot{\mathcal{M}}}{M_{\rm ZC}}R_{\rm ZC}^2\Omega_{\rm ZC}.$$

  Dans le cas d'un gain de masse, $\dot{\mathcal{M}} > 0$, on suppose que l'apport de masse est animé d'une vitesse angulaire égale à celle de la couche externe avec un rayon de giration égal au rayon de l'étoile,$\varpi > 0$ peut être utilisé pour adapter ces dispositions.

– `pertw_0` : On ignore les perte / gain de moment cinétique. La variation temporelle de moment cinétique est modélisée par :

$$\dot{\mathcal{M}}_\Omega = 0.$$

Chutes de planètoîdes : un gain/perte de moment cinétique de la zone convective externe résulte des chutes de planètoïdes, *cf.* § 7.8.2 (Page 99). En notant $\Omega_{\rm pl}$ la vitesse angulaire des planètoïdes, $R_{\rm pl} \geq R_\star$ le rayon de giration des planètoïdes, l'apport de moment cinétique par unité de masse est :

$$\dot{\mathcal{M}}_\Omega = \frac{N_{\rm P} M_\oplus}{M_{\rm ZC}} \mathcal{P}(t) R_{\rm pl}^2 \Omega_{\rm pl}.$$

$N_{\rm P} > 0$ est le nombre total de planètoïdes de masse terrestre, $M_\oplus$, reçus par l'étoile, $\mathcal{P}(t)$ la fonction décrivant la dépendance temporelle, *cf.* § 4.6 (Page 41). Une vitesse angulaire positive des planètoïdes accroit le moment cinétique total de la zone convective externe.

7.10 La convection

7.10.1 Critères de convection

L'énergie est transportée par convection lorsqu'est vérifié *cf.* Cox & Guili [1968] p. 276 ou encore Kippenhahn & Weigert [1991] p. 39 :

– soit le critère de Ledoux, qui tient compte du gradient de composition chimique :

$$\nabla_{\rm rad} < \nabla_{\rm ad}^* + \frac{\varphi}{\delta} \nabla_\mu \sim \frac{4-3\beta}{\beta} (\nabla_{\rm ad}^* - \nabla_{\rm rad}) + \nabla_\mu < 0, \ (\varphi \equiv \frac{d \ln \rho}{d \ln \mu}, \ \nabla_\mu \equiv \frac{d \ln \mu}{d \ln P})$$

ici $\beta = P_r / P$, P_r est la pression de radiation, et :

$$\mu^{-1} = \mu_i^{-1} + \mu_e^{-1} \sim 2X + \frac{3}{4} Y + \frac{Z}{2} = \frac{3 + 5X - Z}{4}, \ \nabla_\mu = \frac{4\pi R^4}{GM} \frac{P}{\mu} \frac{\partial \mu}{\partial X} \frac{\partial X}{\partial M},$$

– soit le critère de Schwarzchild :

$$\nabla_{\rm ad}^* - \nabla_{\rm rad} < 0.$$

On a noté $\nabla_{\rm ad}^*$ le gradient corrigé de la pression turbulente :

$$\nabla_{\rm ad}^* = \frac{\Gamma_2 - 1}{\Gamma_2} \frac{d \ln P_{\rm gaz}}{d \ln P}. \tag{7.83}$$

Le calcul de $\frac{\varphi}{\delta} \nabla_\mu$ nécessite la connaissance des taux d'ionisation partielle des divers éléments. Ces quantités ne sont pas explicitement accessibles avec les équations d'état tabulées dont on dispose. Pour le critère de Ledoux on utilise :

$$\frac{\varphi}{\delta} \nabla_\mu = -\frac{4\pi R^2 H_{\rm p}}{\delta} \sum_{i=1,{\rm nelem}} \frac{\partial X_i}{\partial M}.$$

$\boxed{\textbf{PB}}$ `Bien que dans la présente version de` CESAM2*k* `la somme ne porte que sur l'hydrogène ,l'usage du critère de Ledoux est` **délicat**`.`

Suivant la valeur `.TRUE.` ou `.FALSE.`, de la variable logique `ledoux` de la NAMELIST `NL_CONV` du fichier de données, l'un ou l'autre de ces deux critères est utilisé. Le critère de convection est formulé dans les routines `thermo`, *cf.* §8.128 (Page 207) `thermo_atm`, *cf.* §8.129 (Page 208) et `dgrad`, *cf.* §8.30 (Page 155). Dans la restitution de la partie convective de l'atmosphère, le gradient radiatif sera modifié afin d'assurer la continuité du gradient de température, *cf.* §7.3.3 (Page 78).

7.10.2 Calcul du gradient convectif

La routine `conv_jmj`, *cf.* §8.24 (Page 153), utilise un formalisme proche de celui de Henyey *et al.* [1965] :

$$\nabla - \nabla^*_{\rm ad} = \frac{\Gamma(\Gamma+1)}{B},\ \nabla^*_{\rm ad} = \frac{\Gamma_2 - 1}{\Gamma_2}\frac{d\ln P_{\rm gaz}}{d\ln P},\ B = \xi\frac{l^4 g\delta(\rho c_{\rm p})^2}{H_{\rm p}K^2}, \tag{7.84}$$
$$K = \frac{4acT^3}{3\kappa\rho},\ g = \frac{GM}{R^2},\ \xi = \frac{1}{72}\left(3\frac{V}{Al}\right)^2\left(1 + \frac{2Al}{3V\tau^2}\right)^2,$$

$P_{\rm gaz}$ est la pression gazeuse *i.e.* thermodynamique + radiative et P est la pression totale *i.e.* gazeuse + turbulente, *cf.* §7.10.3 (Page 128), $P = P_{\rm gaz} + P_{\rm tur}$, τ est l'épaisseur optique Rosseland de l'élément convectif :

$$\tau = \kappa\rho l = \kappa\rho\alpha H_{\rm p},$$

$l = \alpha H_{\rm p}$ est la longueur de mélange, $H_{\rm p} = -\frac{dR}{d\ln P} = P/\rho g$ est l'échelle de hauteur de pression. Γ est l'efficacité de la convection, zéro réel de la cubique :

$$\Phi\Gamma^3 + \Gamma(\Gamma+1) = B(\nabla_{\rm rad} - \nabla^*_{\rm ad}), \tag{7.85}$$
$$\Phi = \varphi\left(1 + \frac{2Al}{3V\tau^2}\right)^{-1},\ \varphi = \frac{3}{2}\left(3\frac{V}{Al}\right)^{-1}, \nabla_{\rm rad} = \frac{3}{16\pi acG}\frac{\kappa LP}{mT^4}.$$

La quantité V/A est le rapport du volume/surface de l'élément convectif. Le terme correctif $\frac{\partial\ln P_{\rm gaz}}{\partial\ln P}$ qui affecte le gradient adiabatique $\nabla_{\rm ad} \equiv \frac{\Gamma_2-1}{\Gamma_2}$ dans l'expression de ∇^*, *cf.* Eq. 7.84 (Page 127), est introduit pour avoir une estimation du gradient de température lors d'une transformation adiabatique avec pression turbulente, ce qui est très approximatif et physiquement mal établi.

En un point d'une zone convective, il semble irréaliste que la longueur de mélange puisse être supérieure à la distance qui sépare ce point de la plus proche limite de la zone convective. Ainsi qu'Eggleton [1972] l'a proposé, dans la routine `conv_a0`, *cf.* §8.21 (Page 153), la longueur de mélange est prise égale à $l = \alpha_0 H_{\rm p}$ avec $\alpha_0 \equiv \alpha(1-\nabla^*_{\rm ad}/\nabla_{\rm rad})$, la longueur de mélange utilisée devenant nulle à chaque limite zone radiative / zone convective. Cette approximation est justifiée par le fait que les grandes incertitudes inhérentes à la théorie ne justifient pas d'introduire une trop grande rigueur dans les détails, *cf.* Cox & Guili [1968]. Par ailleurs, l'introduction de la distance exacte change la nature du problème, il devient intégro-différentiel ce qui complique considérablement la résolution numérique. Introduire la distance

exacte à la plus proche limite n'est pas immédiat, du moins algorithmiquement, car le problème devient globalement implicite. Ce qu'évite la formulation locale d'Eggleton.
Dans `conv_jmj` et `conv_a0` ces équations sont résolues avec $\xi = 1/162$, $\Phi = 9/4$ et $V/Al = 2/9$. La recherche du zéro de la cubique est faite à l'aide de l'algorithme de déflation.
La mise en œuvre de `conv_jmj` a bénéficié de la collaboration de J.Provost et de MJ.Goupil.

Dans la routine `conv_cm`, *cf.* § 8.22 (Page 153), le gradient de température dans les zones convectives est calculé selon les prescriptions de Canuto & Mazitelli [1991] ; l'expression (63) de ces auteurs, mise sous la forme implicite équivalente :

$$0 = \nabla + \Phi(\nabla - \nabla^*_{\rm ad}) - \nabla_{\rm rad}$$

est résolue itérativement. Φ est donné par l'expression (32) de Canuto & Mazitelli [1991] :

$$\Phi = a_1\Sigma^m[(1+a_2\Sigma)^n - 1]^p,\ \Sigma = 4A^2(\nabla - \nabla^*_{\rm ad}),\ A = \frac{l^2}{9\chi}\sqrt{\frac{g}{2H_{\rm p}}},\ \chi = \frac{K}{\rho c_{\rm p}}$$

ainsi que (5) et (6) avec $a_1 = 24.868$, $a_2 = 9.7666 \times 10^{-2}$, $m = 0.14972$, $n = 0.18931$ et $p = 1.8503$. Comme définie précédemment, la longueur de mélange est prise égale à $l = \alpha H_{\rm p}$. L'efficacité de la convection est alors l'expression (8) :

$$\Gamma = \frac{\sqrt{\Sigma + 1} - 1}{2}.$$

Dans la routine `conv_cm_reza`, *cf.* § 8.23 (Page 153), similaire à `conv_cm`, il est tenu compte de la quantité thermodynamique $\delta = (\frac{\partial \ln \rho}{\partial \ln T})_P$. Dans la routine `conv_cgm_reza`, *cf.* § 8.23 (Page 153) le calcul du gradient convectif est effectué selon les prescriptions de Canuto *et al.* [1996] avec la prescription de Bernkopf (Heiter *et al.* [2002]) Ces deux routines ont été mises en œuvre et mises à la disposition des utilisateurs de CESAM2k par Reza Samadi, LESIA, Observatoire de Paris

7.10.3 Pression turbulente

La théorie de la pression turbulente est très approximative ; son expression est donnée par la relation phénoménologique :

$$P_{\rm turb} = \rho < v^2 >= a\rho v^2,\ v^2 = \frac{\Gamma}{1+\Gamma}\frac{\delta\alpha^2 P}{8\rho}(\nabla - \nabla^*_{\rm ad}),$$

où v est la "vitesse" des éléments convectifs ; la valeur du coefficient phénoménologique a est mal déterminée, et varie suivant les auteurs, $a \sim 0.21$ pour Canuto & Mazitelli [1991], $a \sim 0.50$ chez Henyey *et al.* [1965] ou encore, $a = \pi/2$.

Dans les conditions de la zone convective solaire, ce n'est que lorsque la température est inférieure à 50 000K que le rapport $P_{\rm turb}/P$ dépasse 1/10 000 ; ce rapport présente un maximum, de l'ordre de 16%, vers 8 000K, *i.e.* à l'intérieur de l'atmosphère. Cette correction de pression turbulente n'est supérieure à 1% que dans une zone localisée sous la surface où les températures se situent dans l'intervalle 6 500K, 15 500K ; ce qui correspond au début de la zone d'ionisation de l'hydrogène.

En tenant compte de la pression turbulente, le problème devient totalement implicite ; en effet, lorsque la pression totale, la température et la composition chimique sont connues, le calcul des grandeurs thermodynamiques nécessite la connaissance de la vitesse de convection, déduite elle même des grandeurs thermodynamiques calculées à partir de la pression gazeuse, donc de la pression totale et de la pression turbulente. Dans CESAM2*k* ce problème implicite est résolu en considérant la pression totale et la pression gazeuse comme deux variables dépendantes distinctes.

La mise en place de la pression turbulente a bénéficié de la collaboration de S. Brun.

PB La correction empirique $\frac{d\ln P_{\rm gaz}}{d\ln P}$ affectant le gradient adiabatique rend le problème extrèmement instable. Dans CESAM2*k* cette correction est ignorée si la donnée CPTURB de la NAMELIST NL_CONV est positive ou nulle.

PB CESAM2*k* permet d'abandonner (*respt.* de réintroduire) la pression turbulente en cours et à *fortiori* au début d'une évolution, le nombre d'équations de structure effectivement résolues est alors 7, (*respt.* 8).

7.10.4 Localisation des limites des zones convectives

Hors atmosphère, cette localisation a pour but essentiel de fixer les limites de la zone à mélanger *i.e.* par mélange convectif, afin de localiser avec précision les discontinuités qui en résultent. Dans l'atmosphère, cette localisation est sans objet puisque la composition chimique et la vitesse angulaires y sont supposées constantes.

Les limites des zones convectives sont déterminées dans la routine `lim_zc`, *cf.* § 8.69 (Page 174). Dans un premier temps on localise en indice de couche, à l'aide du sous-programme `dgrad`, *cf.* § 8.30 (Page 155), les intervalles où se produisent un changement de signe de la quantité $\nabla_{\rm rad} - \nabla^*_{\rm ad}$. Dans une seconde étape, la localisation de chaque changement de signe est affinée, par dichotomie, jusqu'à la précision définie par la variable `loc_zc`. Celle-ci ayant été initialisée dans la routine `cesam`, *cf.* § 8.10 (Page 147), suivant le type de précision requis. Ainsi, avec la précision `sa` *cf.* Table 3.1 (Page 26), la limite de la zone convective solaire sera déterminée à mieux que `loc_zc`$=10^{-5}$ de la largeur de l'intervalle la contenant *e.g.* à mieux que $\sim 30\,$m. C'est cette localisation améliorée qui sert de référence pour d'éventuelles extensions.

7.10.5 Extension des zones convectives

L'extension des zones convectives (`jpz=.FALSE.` dans la NAMELIST `NL_CONV`) et la pénétration convective (`jpz=.TRUE.`) ont pour effet d'étendre la zone de mélange et de fixer le gradient de température égal au gradient adiabatique dans les zones d'extension. Dans CESAM2k on distingue undershooting[21] *i.e.* extension de la limite vers le centre, et overshooting *i.e.* extension de la limite vers la surface ; leur importance est fixée, respectivement, par les paramètres `ovshts` et `ovshti` de la NAMELIST `NL_CONV` du fichier de données ; les zones convectives sont respectivement étendues de `ovshts`$\times H_\mathrm{p}$ et de `ovshti`$\times H_\mathrm{p}$. En ce qui concerne l'overshoot d'un cœur convectif de rayon R_c, l'extension de la zone mélangée est limitée à min(`ovshts`$\times H_\mathrm{p}, R_\mathrm{c}\times$ `ovshts`) si `jpz=.FALSE.` ; elle a pour valeur R_c, si `jpz=.TRUE.` Pour la pénétration convective, obtenue avec `jpz=.TRUE.`, l'extension de la zone mélangée L_p et du gradient adiabatiaque est déterminée par Zahn, [1991] :

$$L_\mathrm{p} = \frac{\zeta H_\mathrm{p}}{K_\mathrm{p}},\ K_\mathrm{p} = \frac{\partial \ln K}{\partial \ln P},$$

$\zeta \sim 0.5=$`ovshti`, étant un paramètre ajustable.
Dans une extension le gradient est pris égal au gradient adiabatique lorsque le paramètre d'overshooting est positif, au gradient radiatif avec une valeur négative, *cf.* § 3.8 (Page 29). La composition chimique et la vitesse angulaire sont homogénéisées, *cf.* § 7.10.6 (Page 130)

Les dispositions suivantes sont prévues :
- La pénétration convective n'a d'action que sur les limites inférieures des zones convectives.
- La pénétration convective et les overshoots n'ont d'effet que si la température est supérieure[22] à 500 000K ; cette disposition a pour but d'éviter "d'overshooter" les zones convectives superficielles.
- Avec la présente version de CESAM2k, on ne peut donc pas étendre une limite zone radiative / zone convective située dans l'atmosphère.

7.10.6 Mélange convectif

Dans les zones mélangées *i.e.* les zones convectives et leurs extensions par pénétration convective, la composition chimique et la vitesse anguliare sont homogénéisées. Avec diffusion on utilise un coefficient de diffusion de mélange turbulent, typiquement, $d_M = 10^{13}$ cm s^{-1}, ordre de grandeur obtenu comme produit de valeurs typiques de la vitesse de convection et de l'échelle de hauteur de pression. En l'absence de diffusion, le mélange est effectué par moyennes spatiales couplées avec les réactions thermonucléaires *i.e.* par la résolution numérique simultanée du mélange et de l'évolution nucléaire, ce qui constitue un problème intégro-différentiel, *cf.* § 8.117 (Page 199).

7.10.7 Estimation de la fréquence de Brunt-Väissälä

Dans les fichiers de sortie en ASCII, CESAM2k donne la valeur de la quantité :

$$A = \frac{1}{\Gamma_1}\frac{\partial \ln P}{\partial \ln R} - \frac{\partial \ln \rho}{\partial \ln R} \tag{7.86}$$

[21] Au sens d'Oxford, undershooting et overshooting sont utilisés de façon incorrecte, ce qui provoque des réactions épidermiques, parfois violentes, de collègues de pure expression anglo-saxonne ; en franglais ils ont le sens imagé d'extension inférieure et supérieure.

[22] Pour modifier cette valeur il faut intervenir dans la routine `lim_zc`.

pour chaque couche du modèle. A permet l'estimation de la fréquence de Brunt-Väissälä ; A, différence de deux dérivées premières, est équivalente à une dérivée seconde. Le calcul de la fréquence de Brunt-Väissälä est donc sensible aux discontinuités de la dérivée première de la composition chimique qui provoquent des oscillations ; on a indiqué au § 6.2 (Page 52), que la prise en compte d'une légère diffusion turbulente permettait d'obtenir des profils quasi théoriques pour la fréquence de Brunt-Väissälä.

Sur une limite zone radiative / zone convective, la composition chimique est une fonction discontinue si on ignore la diffusion microscopique. Avec diffusion, la densité est continue, non dérivable, A se comporte alors comme une fonction δ. Pour décrire finement le comportement de la fréquence de Brunt-Väissälä, dans les fichiers de sortie destinés aux exploitations sismologiques, *cf.* § 8.86 (Page 187), des points sont ajoutés au voisinage de ces discontinuités, *cf.* § 8.42 (Page 161). La dérivée spatiale de la densité étant accessible dans CESAM2*k*, l'équation 7.86 (Page 130) est utilisée pour le calcul de la fréquence de Brunt-Väissälä. Pour que cette relation soit aussi appliquable il est nécessaire que l'approximation de la densité soit numériquement satisfaisante ; ce qui n'est réalisé que lorsque la densité est continue aux limites zones radiatives / zones convectives. ***Cette formulation n'est donc appliquable que si la diffusion des éléments chimiques est prise en compte***. Si tel n'est pas le cas, comme par exemple avec la précision `np` *cf.* Table 3.1 (Page 26) ou sans diffusion.

Dans ces cas une formulation approchée est utilisée ; elle repose sur une approximation de $d\ln\rho$:

$$d\ln\rho = \alpha\, d\ln P - \delta\, d\ln T + \varphi\, d\ln\mu$$

φ n'est donné ni par les routines ni par les tabulations d'équation d'état. L'approximation "gaz parfait" du coefficient thermodynamique :

$$\varphi = \frac{\partial\ln\rho}{\partial\ln\mu} = 1$$

est utilisée. En tenant compte de la relation thermodynamique :

$$\frac{1}{\Gamma_1} = \alpha - \delta\nabla_{\rm ad}$$

on obtient la relation utilisée par CESAM2*k* :

$$A = -\delta\nabla_{\rm ad}\frac{\partial\ln P}{\partial\ln R} + \delta\frac{\partial\ln T}{\partial\ln R} - \varphi\frac{\partial\ln\mu}{\partial\ln R} = \delta\left(\nabla - \nabla_{\rm ad}\right)\frac{\partial\ln P}{\partial\ln R} - \varphi\frac{\partial\ln\mu}{\partial\ln R}. \qquad (7.87)$$

avec, *cf.* § 7.9.4 (Page 112) :

$$\frac{\partial\ln\mu}{\partial\ln R} = -\mu\sum_i(1+\bar{z}_i)\frac{\partial x_i}{\partial\ln R}.$$

Toutefois la formulation utilisée dans les versions précédentes de CESAM reste disponible. Pour la restituer, dans la routine `cesam.f`, *cf.* § 8.10 (Page 147), du sous-directory `SOURCE`, ligne 225, affecter `.FALSE.` au paramètre `new_bv`. L'expression suivante sera utilisée :

$$A = \frac{R\delta}{H_{\rm p}}\left(\nabla_{\rm ad} - \nabla\right) - 4\pi R^3\sum_i\frac{\partial\rho}{\partial X_i}\frac{\partial X_i}{\partial M}. \qquad (7.88)$$

Cette formulation présente des inconvénients :

- Elle nécessite les dérivées de la densité par rapport aux différentes espèces chimiques, quantités inaccessibles avec les équations d'état tabulées, si bien que la somme sur les espèces chimiques est nécessairement réduite au seul élément H (*i.e.* $i \equiv 1$). Une conséquence est que Eq. 7.86 (Page 130) conduit à des valeurs systématiquement inférieures à celles données par Eq. 7.88 (Page 131).
- Un autre inconvénient est la non dérivabilité de la densité aux limites zone radiative/zone convective. Comme d'une part, ces limites ne peuvent coïncider *parfaitement* avec les points de grille et que, d'autre part, les équations d'évolution de la composition chimique ne sont pas résolues simultanément avec les équations de la structure, il se produit nécessairement un petit décalage entre la discontinuité de la composition chimique, donc de la densité (ou seulement de leurs gradients), et le point limite où l'égalité des gradients est réalisée. De ces inconsistances, il résulte des oscillations, parfois violentes des valeurs de A obtenues par Eq. 7.86 (Page 130) et, de là, de la fréquence de Brunt-Väissälä.

Toutefois le gradient de composition chimique étant supposé nul dans l'atmosphère restituée, ces difficultés ne s'y présentent pas et Eq. 7.88 (Page 131) est utilisée.

On présente une autre formulation, qui bien qu'implantée dans CESAM2k, n'est pas utilisée. Comme Eq. 7.86 (Page 130), elle nécessite une oscularité suffisante des variables. La pression P, le rayon, la masse, étant approchés par une spline en fonction de la variable d'indice q, on a accès à leurs dérivées premières et secondes. L'idée est d'en déduire le gradient de densité puisque :

$$\rho = -\frac{1}{g}\frac{\partial P}{\partial R},\ g = \frac{GM}{R^2}.$$

Ainsi on tiendra compte *implicitement* et de façon cohérente, à travers la formulation numérique, des détails de l'équation d'état et des discontinuités. On a :

$$\frac{\partial \ln \rho}{\partial \ln R} = \frac{R}{\rho}\frac{\partial \rho}{\partial R} = \frac{R}{\rho g}\left(\frac{1}{g}\frac{\partial g}{\partial R}\frac{\partial P}{\partial R} - \frac{\partial^2 P}{\partial R^2}\right), \tag{7.89}$$

$$\frac{\partial g}{\partial R} = \frac{G}{R^2}\frac{\partial M}{\partial R} - \frac{2g}{R} = 4\pi\rho G - \frac{2g}{R}. \tag{7.90}$$

A des fins de référence et sous une forme adaptée à la programmation, on donne ci-après les expressions des deux termes de A (Eq. 7.86 (Page 130)) en fonction des quantités ξ, ζ et μ et de leurs dérivées par rapport à la variable d'espacement q.

- Avec les variables $\xi = \ln P$, $\zeta = \left(\frac{R}{R_\odot}\right)^2$ et $\mu = \left(\frac{M}{M_\odot}\right)^{2/3}$:

$$\frac{\partial \zeta}{\partial R} = \frac{2R}{R_\odot^2} = \frac{2\sqrt{\zeta}}{R_\odot},\ \frac{\partial}{\partial R} = \frac{\partial q}{\partial \zeta}\frac{\partial \zeta}{\partial R}\frac{\partial}{\partial q} = \frac{2\sqrt{\zeta}}{R_\odot}\frac{\partial q}{\partial \zeta}\frac{\partial}{\partial q},$$

$$\frac{\partial \ln P}{\partial R} = \frac{\partial \ln P}{\partial q}\frac{\partial q}{\partial \zeta}\frac{\partial \zeta}{\partial R} = \frac{2\sqrt{\zeta}}{R_\odot}\frac{\partial \ln P}{\partial q}\frac{\partial q}{\partial \zeta},$$

$$\frac{\partial \ln P}{\partial \ln R} = R\frac{\partial \ln P}{\partial R} = 2\zeta\left[\frac{\partial \xi}{\partial q}\Big/\frac{\partial \zeta}{\partial q}\right]. \tag{7.91}$$

$$\frac{\partial P}{\partial R}=P\frac{\partial\ln P}{\partial R}=\frac{2RP}{R_\odot^2}\frac{\partial\ln P}{\partial q}\frac{\partial q}{\partial\zeta},\tag{7.92}$$

$$\begin{aligned}\frac{\partial}{\partial R}\frac{\partial q}{\partial\zeta}&=\frac{\partial}{\partial R}\left(\frac{\partial\zeta}{\partial q}\right)^{-1}=-\left(\frac{\partial\zeta}{\partial q}\right)^{-2}\frac{\partial}{\partial R}\frac{\partial\zeta}{\partial q}\\&=-\left(\frac{\partial\zeta}{\partial q}\right)^{-2}\frac{2\sqrt{\zeta}}{R_\odot}\frac{\partial q}{\partial\zeta}\frac{\partial}{\partial q}\frac{\partial\zeta}{\partial q}=-\frac{2\sqrt{\zeta}}{R_\odot}\left(\frac{\partial\zeta}{\partial q}\right)^{-3}\frac{\partial^2\zeta}{\partial q^2},\end{aligned}$$

$$\frac{\partial}{\partial R}\frac{\partial\ln P}{\partial q}=\frac{2\sqrt{\zeta}}{R_\odot}\frac{\partial q}{\partial\zeta}\frac{\partial}{\partial q}\frac{\partial\ln P}{\partial q}=\frac{2\sqrt{\zeta}}{R_\odot}\left(\frac{\partial\zeta}{\partial q}\right)^{-1}\frac{\partial^2\ln P}{\partial q^2},$$

$$\begin{aligned}\frac{\partial^2P}{\partial R^2}&=\frac{\partial}{\partial R}\frac{\partial P}{\partial R}=\frac{\partial}{\partial R}\left\{\frac{2RP}{R_\odot^2}\frac{\partial\ln P}{\partial q}\frac{\partial q}{\partial\zeta}\right\}\\&=\frac{1}{R}\frac{\partial P}{\partial R}+\frac{1}{P}\left(\frac{\partial P}{\partial R}\right)^2+\left[\frac{\partial P}{\partial R}\Big/\frac{\partial q}{\partial\zeta}\right]\frac{\partial}{\partial R}\frac{\partial q}{\partial\zeta}+\left[\frac{\partial P}{\partial R}\Big/\frac{\partial\ln P}{\partial q}\right]\frac{\partial}{\partial R}\frac{\partial\ln P}{\partial q}\\&=\frac{\partial P}{\partial R}\left\{\frac{1}{R}+\frac{1}{P}\frac{\partial P}{\partial R}+\frac{2R}{R_\odot^2}\left[\frac{\partial^2\xi}{\partial q^2}\Big/\frac{\partial\xi}{\partial q}-\frac{\partial^2\zeta}{\partial q^2}\Big/\frac{\partial\zeta}{\partial q}\right]\Big/\frac{\partial\zeta}{\partial q}\right\}\end{aligned}\tag{7.93}$$

L'expression de A donnée par Eq. 7.86 (Page 130) est obtenue par Eq. 7.91 (Page 132) et Eq. 7.89 (Page 132) calculée à l'aide de Eq. 7.90 (Page 132), Eq. 7.92 (Page 133) et Eq. 7.93 (Page 133).

– Avec les variables $\xi=\ln P$, $\zeta=\frac{R}{R_\odot}$ et $\mu=\frac{M}{M_\odot}$:

$$\frac{\partial\zeta}{\partial R}=\frac{1}{R_\odot},\ \frac{\partial}{\partial R}=\frac{\partial q}{\partial\zeta}\frac{\partial\zeta}{\partial R}\frac{\partial}{\partial q}=\frac{1}{R_\odot}\frac{\partial q}{\partial\zeta}\frac{\partial}{\partial q},$$

$$\frac{\partial\ln P}{\partial R}=\frac{\partial\ln P}{\partial q}\frac{\partial q}{\partial\zeta}\frac{\partial\zeta}{\partial R}=\frac{1}{R_\odot}\frac{\partial\ln P}{\partial q}\frac{\partial q}{\partial\zeta},$$

$$\frac{\partial\ln P}{\partial\ln R}=R\frac{\partial\ln P}{\partial R}=\frac{R}{R_\odot}\left[\frac{\partial\xi}{\partial q}\Big/\frac{\partial\zeta}{\partial q}\right].\tag{7.94}$$

$$\frac{\partial P}{\partial R}=P\frac{\partial\ln P}{\partial R}=\frac{P}{R_\odot}\frac{\partial\ln P}{\partial q}\frac{\partial q}{\partial\zeta},\tag{7.95}$$

$$\begin{aligned}\frac{\partial}{\partial R}\frac{\partial q}{\partial\zeta}&=\frac{\partial}{\partial R}\left(\frac{\partial\zeta}{\partial q}\right)^{-1}=-\left(\frac{\partial\zeta}{\partial q}\right)^{-2}\frac{\partial}{\partial R}\frac{\partial\zeta}{\partial q}\\&=-\left(\frac{\partial\zeta}{\partial q}\right)^{-2}\frac{1}{R_\odot}\frac{\partial q}{\partial\zeta}\frac{\partial}{\partial q}\frac{\partial\zeta}{\partial q}=-\frac{1}{R_\odot}\left(\frac{\partial\zeta}{\partial q}\right)^{-3}\frac{\partial^2\zeta}{\partial q^2},\end{aligned}$$

$$\frac{\partial}{\partial R}\frac{\partial\ln P}{\partial q}=\frac{1}{R_\odot}\frac{\partial q}{\partial\zeta}\frac{\partial}{\partial q}\frac{\partial\ln P}{\partial q}=\frac{1}{R_\odot}\left(\frac{\partial\zeta}{\partial q}\right)^{-1}\frac{\partial^2\ln P}{\partial q^2},$$

$$\begin{aligned}\frac{\partial^2P}{\partial R^2}&=\frac{\partial}{\partial R}\frac{\partial P}{\partial R}=\frac{\partial}{\partial R}\left\{\frac{P}{R_\odot}\frac{\partial\ln P}{\partial q}\frac{\partial q}{\partial\zeta}\right\}\\&=\frac{1}{P}\left(\frac{\partial P}{\partial R}\right)^2+\left[\frac{\partial P}{\partial R}\Big/\frac{\partial q}{\partial\zeta}\right]\frac{\partial}{\partial R}\frac{\partial q}{\partial\zeta}+\left[\frac{\partial P}{\partial R}\Big/\frac{\partial\ln P}{\partial q}\right]\frac{\partial}{\partial R}\frac{\partial\ln P}{\partial q}\\&=\frac{\partial P}{\partial R}\left\{\frac{1}{P}\frac{\partial P}{\partial R}+\frac{1}{R_\odot}\left[\frac{\partial^2\xi}{\partial q^2}\Big/\frac{\partial\xi}{\partial q}-\frac{\partial^2\zeta}{\partial q^2}\Big/\frac{\partial\zeta}{\partial q}\right]\Big/\frac{\partial\zeta}{\partial q}\right\}.\end{aligned}\tag{7.96}$$

L'expression de A donnée par Eq. 7.86 (Page 130), est obtenue par Eq. 7.94 (Page 133) et Eq. 7.89 (Page 132) calculée à l'aide de Eq. 7.90 (Page 132), Eq. 7.95 (Page 133) et Eq. 7.96 (Page 133).

PB `Ces dernières relations doivent être adaptées si on tient compte de la rotation.`

PB `Des tests héliosismologiques ont mis en évidence que ces relations, bien que formellement plus cohérentes, faisaient ressortir les incohérences entre équation d'état et opacités. Il convient de ne les utiliser qu'avec prudence.`

7.11 Les réactions thermonucléaires

7.11.1 Abondances initiales

Pour éviter des calculs inutiles, il est nécessaire d'optimiser le réseau nucléaire utilisé en fonction de la connaissance, à priori, de la physique des modèles à calculer. Il est par exemple inutile de tenir compte des réactions thermonucléaires concernant le silicium si la température maximum n'atteint pas le seuil du déclanchement de ces réactions. Il est donc inutile de suivre explicitement l'évolution des éléments chimiques qui ne sont pas concernés par le réseau nucléaire retenu, sauf, évidemment, si on désire en suivre la diffusion. Les isotopes de la mixture initiale n'étant pas retenus comme éléments chimiques ils ne doivent pas pour autant être totalement ignorés, d'autant que leur contribution à la masse totale, quelques 10^{-3}, est loin d'être négligeable. Dans CESAM2k, ces éléments sont représentés par l'isotope *fictif* dont la masse (*respt.* charge) égale la moyenne pondérée des masses (*respt.* charges) des éléments dont il n'est pas tenu compte explicitement. Par exemple cet isotope fictif, ici noté Ex, est souvent ^{24}Si avec une mixture initiale solaire. Cet isotope fictif complémentaire a l'indice `i_ex`, *cf.* §8.2 (Page 141).

L'abondance initiale en *nombre par mole* de chaque isotope de la mixture utilisée dans CESAM2k est déterminée en fonction du réseau de réactions retenu, *cf.* §8.3 (Page 143), à partir :

- Des abondances initiales *en masse* de l'hydrogène X, et de l'hélium Y, lues dans la NAMELIST `NL_CHIM` du fichier de données, *cf.* §3.3 (Page 22).
- Des proportions initiales *en nombre* des éléments lourds dans $Z = 1 - X - Y$.
- Des rapports isotopiques initiaux.

De l'ensemble de ces données résulte un système d'équations linéaires dont la solution est la composition chimique initiale. Dans l'exemple qui suit, x_i désigne l'abondance par mole *i.e.* l'abondance par masse divisée par la masse atomique ν_i, de l'isotope d'ordre $i = 1, \dots, 10$ dans la liste : ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O, Ex. Les rapports d'abondances initiaux en nombre, dans Z, du carbone, azote et oxygène seront respectivement désignés par C, N et O ; par exemple pour le soleil : $C = 0.24551$, $N = 0.06458$, $O = 0.51295$. Les rapports isotopiques initiaux en nombre seront désignés par le rapport de leurs symboles *e.g.* ^{3}He/^{4}He$= 4.2\,10^{-4}$. Le système d'équations linéaires vérifiées par les abondances initiales en nombre, $x_{i,\ i=1,\dots,10}$ s'écrit :

$$x_1\nu_1 = X$$

$$
\begin{aligned}
x_2\nu_2 + x_3\nu_3 &= Y \\
\Sigma_{i=4}^{10} x_i\nu_i &= 1 - X - Y \\
x_4 + x_5 - C \times \Sigma_{i=4}^{10} x_i &= 0 \\
x_6 + x_7 - N \times \Sigma_{i=4}^{10} x_i &= 0 \\
x_8 + x_9 - O \times \Sigma_{i=4}^{10} x_i &= 0 \\
x_2 - x_3\ {}^3\mathrm{He}/{}^4\mathrm{He} &= 0 \\
x_5 - x_4\ {}^{13}\mathrm{C}/{}^{12}\mathrm{C} &= 0 \\
x_7 - x_6\ {}^{15}\mathrm{N}/{}^{14}\mathrm{N} &= 0 \\
x_9 - x_8\ {}^{17}\mathrm{O}/{}^{16}\mathrm{O} &= 0
\end{aligned}
\tag{7.97}
$$

Pour chaque isotope, on fixe une abondance minimale au dessous de laquelle l'isotope est considéré comme inexistant, son évolution temporelle n'est alors pas contrôlée. La plupart des rapports isotopiques utilisés sont ceux de la nébuleuse solaire issus de la table 3 des nuclides de l'article de Anders et Grevesse [1989]. Le rapport isotopique ${}^3\mathrm{He}/{}^4\mathrm{He}= 1.104 \times 10^{-4}$ est déduit d'une mesure dans Jupiter ; toutefois si ${}^2\mathrm{H}$ est pris à l'équilibre, il convient d'ajouter à l'abondance initiale de ${}^3\mathrm{He}$ celle de ${}^2\mathrm{H}$ qui est transformée en ${}^3\mathrm{He}$ pendant la pré-séquence principale, d'où la valeur du rapport isotopique initial ${}^3\mathrm{He}/{}^4\mathrm{He}= 4.2\,10^{-4}$.

L'ordre des éléments chimiques est en général celui des masses croissantes, l'isotope fictif venant en dernier, mais ce n'est pas une règle intangible. L'ordre est fixé dans la routine `tabul_nuc`, *cf.* § 8.124 (Page 205), *e.g.* dans `ppcno3a9` l'ordre des éléments est : ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O, Ex. L'isotope fictif venant en dernier.

7.11.2 Cycle PP simplifié

Il s'agit de la routine `nuc_gong`, *cf.* § 8.96 (Page 191), du Solar Model Comparison Project de J.Christensen–Dalsgaard [1988]. On ne tient compte que d'un seul élément chimique : l'hydrogène dont l'abondance par unité de masse est ici notée X. L'énergie thermonucléaire générée par unité de masse et de temps est :

$$\epsilon = Q_e a_{11} \frac{X^2}{2m_u} \rho T_9^{-\frac{2}{3}} \exp(-bT_9^{-\frac{1}{3}})$$

qui correspond à un taux :

$$\dot{X} = \frac{\partial X}{\partial t} = -R_e a_{11} \frac{X^2}{2} \rho T_9^{-\frac{2}{3}} \exp(-bT_9^{-\frac{1}{3}})$$

où $R_e = 6.5$, $Q_e = 6.5 \times 10^{-5}$, $a_{11} = 4.21 \times 10^{-15}$, $b = 3.6$, m_u est l'unité de masse atomique et $T_9 = T \times 10^{-9}$.

7.11.3 Exemple de réseau nucléaire : cycles PP, CNO et 3α

A titre d'exemple, on détaille la construction d'un réseau nucléaire rassemblant les cycles PP, CNO et 3α

Pour simplifier les notations, on désigne indistinctement par un même symbole l'isotope et son abondance par mole. Les principales réactions thermonucléaires des cycles PP, CNO et 3α sont les suivantes, Clayton [1968], p. 380 et p. 392 :

$$\underbrace{\begin{array}{ll} R_1 : & {}^1\mathrm{H}(p,\beta^+\nu)^2\mathrm{H} \\ R_2 : & {}^2\mathrm{H}(p,\gamma)^3\mathrm{He} \\ R_3 : & {}^3\mathrm{He}(^3\mathrm{He},2p)^4\mathrm{He} \\ R_4 : & {}^4\mathrm{He}(^3\mathrm{He},\gamma)^7\mathrm{Be} \\ R_5 : & {}^7\mathrm{Li}(p,\alpha)^4\mathrm{He} \\ R_6 : & {}^7\mathrm{Be}(e^-\nu\gamma)^7\mathrm{Li} \\ R_7 : & {}^7\mathrm{Be}(p,\gamma)^8\mathrm{B}^*(\beta^+\nu)^8\mathrm{Be}(\alpha)^4\mathrm{He} \end{array}}_{\text{cycle PP}} \qquad \underbrace{\begin{array}{ll} R_8 : & {}^{12}\mathrm{C}(p,\gamma)^{13}\mathrm{N}(\beta^+\nu)^{13}\mathrm{C} \\ R_9 : & {}^{13}\mathrm{C}(p,\gamma)^{14}\mathrm{N} \\ R_{10} : & {}^{14}\mathrm{N}(p,\gamma)^{15}\mathrm{O}(\beta^+\nu)^{15}\mathrm{N} \\ R_{11} : & {}^{15}\mathrm{N}(p,\gamma)^{16}\mathrm{O} \\ R_{12} : & {}^{15}\mathrm{N}(p,\alpha)^{12}\mathrm{C} \\ R_{13} : & {}^{16}\mathrm{O}(p,\gamma)^{17}\mathrm{F}(\beta^+\nu)^{17}\mathrm{O} \\ R_{14} : & {}^{17}\mathrm{O}(p,\alpha)^{14}\mathrm{N} \end{array}}_{\text{cycle CNO}}$$

$$\underbrace{\begin{array}{ll} R_{15} : & {}^4\mathrm{He}(2\alpha,\gamma)^{12}\mathrm{C} \\ R_{16} : & {}^{12}\mathrm{C}(\alpha,\gamma)^{16}\mathrm{O} \\ R_{17} : & {}^{16}\mathrm{O}(\alpha,\gamma)^{20}\mathrm{Ne} \end{array}}_{\text{cycle } 3\alpha}$$

Compte non tenu de l'effet d'écran, le nombre R_i de réactions i, $i = 1, \dots, 17$, par sec^{-1} g^{-1} est obtenu en mutipliant le taux tabulé par :
- ρ pour les réactions binaires entre éléments différents,
- $\rho/2!$ pour les réactions binaires entre éléments identiques : R_1 et R_3,
- $\rho^2/3!$ pour la réaction ternaire R_{15}, Fowler *et al.* [1975], p. 86 et 87.

Le nombre d'électrons par mole e^- et les taux de réaction incluant l'effet d'écran sont des fonctions $R_{i,\,i=1,\dots,17}(T, \rho, \mathcal{X})$, $\mathcal{X}$ désignant le vecteur de composition chimique. Ils sont calculés ainsi que leur dérivées, par rapport à $(T, \rho, \mathcal{X})$, dans la routine `rq_reac`, en utilisant les tables construites par le programme `tab_reac`.

Bien que les réactions nucléaires ne soient pas uniquement localisées dans un milieu totalement ionisé, *e.g.* le fer n'est pas totalement ionisé au centre du soleil, le nombre d'électrons libres par mole est pris égal à :

$$n_\mathrm{e} = \sum_i z_i x_i.$$

7.11.4 Eléments à l'équilibre

Dans certaines phases de l'évolution, des éléments *e.g.* ^{2}H, ^{7}Li et ^{7}Be pour le soleil, ont des temps caractéristiques d'évolution très courts par rapport à l'échelle de temps de l'évolution. Leur taux de création est égal à leur taux de destruction. De tels éléments sont dits "***à l'équilibre***". L'abondance de chaque élément à l'équilibre résulte directement de la relation d'équilibre, ce qui permet des simplifications importantes. ^{2}H, ^{7}Li et ^{7}Be sont considérés à l'équilibre dans la routine `ppcno3a9`. On a alors :

$$0 = \frac{d\,{}^2\mathrm{H}}{dt} = R_1\,{}^1\mathrm{H}^2 - R_2\,{}^1\mathrm{H}\,{}^2\mathrm{H} \Rightarrow {}^2\mathrm{H} = \frac{R_1\,{}^1\mathrm{H}}{R_2},$$

$$\left\{\begin{array}{l} 0 = \frac{d\ ^7\mathrm{Be}}{dt} = R_4\ ^3\mathrm{He}\ ^4\mathrm{He} - R_6\ ^7\mathrm{Be}\ e^- - R_7\ ^1\mathrm{H}\ ^7\mathrm{Be} \\ \\ 0 = \frac{d\ ^7\mathrm{Li}}{dt} = R_6\ ^7\mathrm{Be}\ e^- - R_5\ ^1\mathrm{H}\ ^7\mathrm{Li} \end{array}\right. \Rightarrow \left\{\begin{array}{l} ^7\mathrm{Be} = \frac{R_4\,^3\mathrm{He}\,^4\mathrm{He}}{R_6\,e^- + R_7\,^1\mathrm{H}} \\ \\ ^7\mathrm{Li} = \frac{R_6\,^7\mathrm{Be}\,e^-}{R_5\,^1\mathrm{H}} \\ \\ R_4\,^3\mathrm{He}\,^4\mathrm{He} = R_5\,^1\mathrm{H}\,^7\mathrm{Li} \\ +R_7\,^1\mathrm{H}\,^7\mathrm{Be} \end{array}\right.$$

7.11.5 Effet d'écran

L'effet d'écran est un abaissement de la barrière de potentiel coulombienne dû à la charge d'espace des ions. Il se traduit par l'augmentation des taux des réactions thermonucléaires. Pour une évolution jusque sur la branche des géantes, l'approximation "*écran faible*" est suffisante, son expression est la suivante : Clayton, [1968], équations 4–215 à 4–221 :

$$f = \exp(-\frac{U_0}{kT}) = \exp\left(0.188 z_1 z_2 \sqrt{\frac{\rho\ \zeta}{T_6^3}}\right), \qquad \zeta = \sum_i z_i(1+z_i)x_i$$

z_i étant la charge du noyau i, x_i l'abondance, par mole, de l'élément chimique i et $T_6 = T \times 10^{-6}$. ζ représente l'effet des charges. L'approximation "*écran faible*", n'est valable que si f est proche de l'unité.

CESAM2*k* offre l'alternative d'utiliser l'écrantage de Mitler [1977], qui généralise, d'après cet auteur, les cas d'écrantage faible et intermédiaire ; le cas écrantage fort n'est pas implémenté dans la version actuelle de CESAM2*k*. Pour son emploi, coder `mitler=.TRUE.` dans la NAMELIST `NL_NUC` du fichier de données, *cf.* § 3.3 (Page 22). L'installation de l'écrantage de Mitler a bénéficié d'une collaboration avec S. Turck-Chièze.

PB `Pour la capture électronique de la réaction` ^{7}Be($e^-\nu\gamma$)^{7}Li`, l'effet d'écran est inclus dans l'expression du taux de réaction (Fowler` *et al.* `[1975]).`

7.11.6 Energie thermonucléaire et neutrinos

La quantité d'énergie thermonucléaire, $Q_i, i = 1, \ldots, 17$, libérée par chaque réaction i, est déterminée pour chaque réaction par le programme `tab_reac` à partir des excès de masse et des énergies des neutrinos, Clayton [1968] p. 289. Ces quantités sont tabulées avec les taux des réactions.

La quantité d'énergie thermonucléaire libérée, est alors :$\sum_i Q_i R_i$ *e.g.* pour les cycles PP, CNO et 3 α, on a :

$$\begin{aligned} \epsilon = {} & q_1\ ^1\mathrm{H}^2 + q_2\ ^1\mathrm{H}\ ^2\mathrm{H} + q_3\ ^3\mathrm{He}^2 + q_4\ ^3\mathrm{He}\ ^4\mathrm{He} + q_5\ ^1\mathrm{H}\ ^7\mathrm{Li} + (q_6\ e^- + q_7\ ^1\mathrm{H})\ ^7\mathrm{Be} \\ & + q_8\ ^1\mathrm{H}\ ^{12}\mathrm{C} + q_9\ ^1\mathrm{H}\ ^{13}\mathrm{C} + q_{10}\ ^1\mathrm{H}\ ^{14}\mathrm{N} + (q_{11}+q_{12})^1\mathrm{H}\ ^{15}\mathrm{N} + q_{13}\ ^1\mathrm{H}\ ^{16}\mathrm{O} + q_{14}\ ^1\mathrm{H}\ ^{17}\mathrm{O} \\ & + q_{15}\ ^4\mathrm{He}^3 + q_{16}\ ^4\mathrm{He}\ ^{12}\mathrm{C} + q_{17}\ ^4\mathrm{He}\ ^{16}\mathrm{O}, \end{aligned}$$

chacune des lignes représente l'énergie libérée respectivement par les cycles PP, CNO et 3α. Ces quantités apparaissent séparément dans la liste du modèle. Avec ^{2}H, ^{7}Li et ^{7}Be à l'équilibre, il convient d'utiliser les valeurs à l'équilibre établies précédemment.

Le nombre de neutrinos, par unité de masse et de temps, émis respectivement par les réactions R_1, R_6, R_7, R_8, R_{10} et R_{13} est :

$$H_n = R_1\ {}^1\mathrm{H}^2,\ Be_n = R_6 e^-\ {}^7\mathrm{Be},\ B_n = R_7\ {}^1\mathrm{H}\ {}^7\mathrm{Li},$$
$$N_n = R_8\ {}^1\mathrm{H}\ {}^{12}\mathrm{C},\ O_n = R_{10}\ {}^1\mathrm{H}\ {}^{14}\mathrm{N},\ F_n = R_{13}\ {}^1\mathrm{H}\ {}^{16}\mathrm{O}.$$

Le nombre de neutrinos reçus sur terre par les expériences du chlore et du galium est calculé en supposant la source distante d' une unité astronomique

7.11.7 Equations d'évolution

Pour l'exemple des cycles PP+CNO+3α, avec ^{2}H, ^{7}Be, et ^{7}Li à l'équilibre, en désignant les abondances par mole par les symboles isotopiques, les équations d'évolution sont les suivantes :

$$\begin{aligned}
\frac{d^1\mathrm{H}}{dt} &= -3R_1\ {}^1\mathrm{H}^2 + 2R_3\ {}^3\mathrm{He}^2 - R_4\ {}^3\mathrm{He}\ {}^4\mathrm{He} \\
&\quad -R_8\ {}^1\mathrm{H}\ {}^{12}\mathrm{C} - R_9\ {}^1\mathrm{H}\ {}^{13}\mathrm{C} - R_{10}\ {}^1\mathrm{H}\ {}^{14}\mathrm{N} - (R_{11}+R_{12}){}^1\mathrm{H}\ {}^{15}\mathrm{N} \\
&\quad -R_{13}\ {}^1\mathrm{H}\ {}^{16}\mathrm{O} - R_{14}\ {}^1\mathrm{H}\ {}^{17}\mathrm{O}, \\
\frac{d^3\mathrm{He}}{dt} &= R_1\ {}^1\mathrm{H}^2 - 2R_3\ {}^3\mathrm{He}^2 - R_4\ {}^3\mathrm{He}\ {}^4\mathrm{He}, \\
\frac{d^4\mathrm{He}}{dt} &= R_3\ {}^3\mathrm{He}^2 + R_4\ {}^3\mathrm{He}\ {}^4\mathrm{He} \\
&\quad +R_{12}\ {}^1\mathrm{H}\ {}^{15}\mathrm{N} + R_{14}\ {}^1\mathrm{H}\ {}^{17}\mathrm{O} \\
&\quad -3R_{15}\ {}^4\mathrm{He}^3 - R_{16}\ {}^4\mathrm{He}\ {}^{12}\mathrm{C} - R_{17}\ {}^4\mathrm{He}\ {}^{16}\mathrm{O}, \\
\frac{d^{12}\mathrm{C}}{dt} &= -R_8\ {}^1\mathrm{H}\ {}^{12}\mathrm{C} + R_{12}\ {}^1\mathrm{H}\ {}^{15}\mathrm{N} \\
&\quad +R_{15}\ {}^4\mathrm{He}^3 - R_{16}\ {}^4\mathrm{He}\ {}^{12}\mathrm{C}, \\
\frac{d^{13}\mathrm{C}}{dt} &= R_8\ {}^1\mathrm{H}\ {}^{12}\mathrm{C} - R_9\ {}^1\mathrm{H}\ {}^{13}\mathrm{C}, \qquad (7.98) \\
\frac{d^{14}\mathrm{N}}{dt} &= R_9\ {}^1\mathrm{H}\ {}^{13}\mathrm{C} - R_{10}\ {}^1\mathrm{H}\ {}^{14}\mathrm{N} + R_{14}\ {}^1\mathrm{H}\ {}^{17}\mathrm{O}, \\
\frac{d^{15}\mathrm{N}}{dt} &= R_{10}\ {}^1\mathrm{H}\ {}^{14}\mathrm{N} - (R_{11}+R_{12}){}^1\mathrm{H}\ {}^{15}\mathrm{N}, \\
\frac{d^{16}\mathrm{O}}{dt} &= R_{11}\ {}^1\mathrm{H}\ {}^{15}\mathrm{N} - R_{13}\ {}^1\mathrm{H}\ {}^{16}\mathrm{O} \\
&\quad +R_{16}\ {}^4\mathrm{He}\ {}^{12}\mathrm{C} - R_{17}\ {}^4\mathrm{He}\ {}^{16}\mathrm{O}, \\
\frac{d^{17}\mathrm{O}}{dt} &= R_{13}\ {}^1\mathrm{H}\ {}^{16}\mathrm{O} - R_{14}\ {}^1\mathrm{H}\ {}^{17}\mathrm{O}, \\
\frac{d\,\mathrm{Ex}}{\mathrm{dt}} &= -\frac{1}{m_{\mathrm{Ex}}}\left[m_{^1\mathrm{H}}\frac{d^1\mathrm{H}}{dt} + m_{^3\mathrm{He}}\frac{d^3\mathrm{He}}{dt} + m_{^4\mathrm{He}}\frac{d^4\mathrm{He}}{dt} + m_{^{12}\mathrm{C}}\frac{d^{12}\mathrm{C}}{dt} + m_{^{13}\mathrm{C}}\frac{d^{13}\mathrm{C}}{dt}\right. \\
&\quad \left. +m_{^{14}\mathrm{N}}\frac{d^{14}\mathrm{N}}{dt} + m_{^{15}\mathrm{N}}\frac{d^{15}\mathrm{N}}{dt} + m_{^{16}\mathrm{O}}\frac{d^{16}\mathrm{O}}{dt} + m_{^{17}\mathrm{O}}\frac{d^{17}\mathrm{O}}{dt}\right],
\end{aligned}$$

la dernière équation exprime la conservation du nombre de baryons, elle permet de tenir compte des éléments qui, comme le ^{20}Ne, n'est brulé par aucune réaction, mais qui est crée, m_i est le nombre atomique de l'espèce chimique i.

Sans élément à l'équilibre, il faut remplacer les trois premières et la dernière équations par :

$$
\begin{aligned}
\frac{d^1\mathrm{H}}{dt} &= -2R_1\ {}^1\mathrm{H}^{\ 2} - R_2\ {}^1\mathrm{H}\ {}^2\mathrm{H} + 2R_3\ {}^3\mathrm{He}^2 - R_5\ {}^1\mathrm{H}\ {}^7\mathrm{Li} - R_7\ {}^1\mathrm{H}\ {}^7\mathrm{Be}\\
&\quad -R_8\ {}^1\mathrm{H}\ {}^{12}\mathrm{C} - R_9\ {}^1\mathrm{H}\ {}^{13}\mathrm{C} - R_{10}\ {}^1\mathrm{H}\ {}^{14}\mathrm{N}\\
&\quad -(R_{11}+R_{12}){}^1\mathrm{H}\ {}^{15}\mathrm{N} - R_{13}\ {}^1\mathrm{H}\ {}^{16}\mathrm{O} - R_{14}\ {}^1\mathrm{H}\ {}^{17}\mathrm{O},\\
\frac{d^2\mathrm{H}}{dt} &= R_1\ {}^1\mathrm{H}^{\ 2} - R_2\ {}^1\mathrm{H}\ {}^2\mathrm{H},\\
\frac{d^3\mathrm{He}}{dt} &= R_2\ {}^1\mathrm{H}\ {}^2\mathrm{H} - 2R_3\ {}^3\mathrm{He}^2 - R_4\ {}^3\mathrm{He}\ {}^4\mathrm{He},\\
\frac{d^4\mathrm{He}}{dt} &= R_3\ {}^3\mathrm{He}^2 - R_4\ {}^3\mathrm{He}\ {}^4\mathrm{He} + 2R_5\ {}^1\mathrm{H}\ {}^7\mathrm{Li} + 2R_7\ {}^1\mathrm{H}\ {}^7\mathrm{Be}\\
&\quad +R_{12}\ {}^1\mathrm{H}\ {}^{15}\mathrm{N} + R_{14}\ {}^1\mathrm{H}\ {}^{17}\mathrm{O}\\
&\quad -3R_{15}\ {}^4\mathrm{He}^3 - R_{16}\ {}^4\mathrm{He}\ {}^{12}\mathrm{C} - R_{17}\ {}^4\mathrm{He}\ {}^{16}\mathrm{O},\\
\frac{d^7\mathrm{Li}}{dt} &= -R_5\ {}^1\mathrm{H}\ {}^7\mathrm{Li} + R_6\ {}^7\mathrm{Be}\ e^-,\\
\frac{d^7\mathrm{Be}}{dt} &= R_4\ {}^3\mathrm{He}\ {}^4\mathrm{He} - R_6\ {}^7\mathrm{Be}\ e^- - R_7\ {}^1\mathrm{H}\ {}^7\mathrm{Be}.\\
\frac{d\,\mathrm{Ex}}{\mathrm{dt}} &= -\frac{1}{m_{\mathrm{Ex}}}\left[m_{^1\mathrm{H}}\frac{d^1\mathrm{H}}{dt} + m_{^2\mathrm{H}}\frac{d^2\mathrm{H}}{dt} + m_{^3\mathrm{He}}\frac{d^3\mathrm{He}}{dt} + m_{^4\mathrm{He}}\frac{d^4\mathrm{He}}{dt}\right.\\
&\quad +m_{^7\mathrm{Li}}\frac{d^7\mathrm{Li}}{dt} + m_{^7\mathrm{Be}}\frac{d^7\mathrm{Be}}{dt} + m_{^{12}\mathrm{C}}\frac{d^{12}\mathrm{C}}{dt} + m_{^{13}\mathrm{C}}\frac{d^{13}\mathrm{C}}{dt}\\
&\quad \left. +m_{^{14}\mathrm{N}}\frac{d^{14}\mathrm{N}}{dt} + m_{^{15}\mathrm{N}}\frac{d^{15}\mathrm{N}}{dt} + m_{^{16}\mathrm{O}}\frac{d^{16}\mathrm{O}}{dt} + m_{^{17}\mathrm{O}}\frac{d^{17}\mathrm{O}}{dt}\right].
\end{aligned}
$$

On dénommera par "jacobien" la matrice des dérivées partielles des équations d'évolution par rapport aux abondances :

$$J(i,j) \equiv \left(\frac{\partial \frac{dx_i}{dt}}{\partial x_j}\right)_{i,j}.$$

Cette matrice est la base de la résolution des équations implicites du problème différentiel raide, *cf.* § 7.5.1 (Page 85), constitué par le réseau nucléaire. Il est nécessaire d'en tenir compte, du fait que l'effet d'écran est fonction des abondances. Une erreur dans l'expression du jacobien entraîne des difficultés de convergence de l'algorithme d'intégration numérique des équations d'évolution. Le programme `test_nuc` du sous-directory `TESTS` peut servir de canevas pour construire un programme permettant de vérifier l'exactitude du jacobien *cf.* § 7.1.4 (Page 63).

Chapitre 8

Les routines de physique

Les grands diseux ne sont pas les grands faiseux.

Proverbe populaire.

La mise en œuvre de l'ensemble de ces routines a bénéficié de nombreuses collaborations avec, en particulier, J.Provost, G.Berthomieu et B.Pichon.

On fait d'abord mention des règles de programmation utilisées dans CESAM2k, puis on décrit l'aspect programmation des routines de physique regroupées dans les modules du sous-directory `SOURCE`. La plupart d'entre elles ne constituent que des applications au problème de la structure interne des méthodes numériques développées au chapitre 7 (Page 59). La description des routines purement numériques fait l'objet du chapitre 9 (Page 213).

Dans le but de faciliter l'exploitation de cette notice, les routines sont décrites en suivant l'ordre alphabétique de leur dénomination.

8.1 Routines génériques

La physique retenue est introduite par l'intermédiaire des mots clefs du fichier de données `mon_modele.don`, *cf.* § 3.15 (Page 33). A chaque mot clef est associée une ***routine générique***, à laquelle sera affectée la ***routine dépendante*** où sera effectué le calcul. Par exemple, dans le fichier de données du § 3.3 (Page 22) `ppcno9` est affecté au mot clef `nom_nuc`. Ce mot clef associe à la routine générique `nuc` la fonction de la routine `ppcno9`. Ainsi, à chaque appel de `CALL nuc(....)`, le calcul des réactions thermonucléaires sera effectué par `ppcno9`, *cf.* § 8.110 (Page 194).

8.2 Unités

Les arguments des routines sont en ***unité cgs SAUF*** : la masse, la luminosité et le rayon qui sont en unités solaires *i.e.* $M_\odot$,$R_\odot$ et $L_\odot$. Si, dans un sous-programme, l'évaluation de certaines variables de sortie fait intervenir des variables d'entrée dont les unités sont dans des systèmes différents, la cohérence entre les unités des variables d'entrée et de sortie est respectée. Par exemple, dans le sous-programme `thermo`, *cf.* § 8.128 (Page 207), qui calcule diverses grandeurs et leurs dérivées premières à partir des variables thermodynamiques, la variable `gradient` ($\nabla = \partial \ln T / \partial \ln P$) est sans dimension. Le calcul de la variable `dgradm`=($\partial\nabla/\partial m$) nécessite la variable d'entrée `m` (masse) qui est sans dimension : $[m/M_\odot]$ ainsi, `dgradm` est sans dimension et non pas en g^{-1}.

Les noms des variables sont choisis de façon à être facilement identifiables.
La composition chimique est un argument d'entrée de divers sous-programmes. Les proportions par mole – proportion par gramme / masse atomique – des divers éléments sont les coordonnées d'un vecteur très souvent appelé : `xchim` où les isotopes ***effectivement utilisés*** sont souvent rangés dans l'ordre des masses atomiques croissantes, ordre qui est défini dans le sous-programme `tabul_nuc`, *cf.* § 8.124 (Page 205), lors de la création des tables de réactions nucléaires, *cf.* § 7.1.4 (Page 63). Cet ordre n'est pas impératif, sauf en ce qui concerne ^{1}H qui doit être le premier isotope.
Des valeurs spécifiques sont utilisées pour :

– Indice de ^{1}H : 1.
– Indice de ^{4}He : `ihe4`
– Indice de ^{3}He : `ihe4-1`
– Indice de ^{56}Fe : `ife56`.
– Indice de l'élément fictif complémentaire : `i_ex`, *cf.* § 7.11.1 (Page 134).

Les variables sont toutes de ***type déclaré***, sauf pour certains programmes de source externe, exemple : `z14xcotrin21` *cf.* § 8.136 (Page 212). La plupart des constantes physiques fondamentales utilisées par CESAM2k sont réunies dans le module `mod_donnees`, *cf.* § E.3 (Page 248). Elles sont initialisées au début de chaque exécution par un appel au sous-programme `ini_ctes`, *cf.* § 8.61 (Page 171), au début du sous-programme `cesam`, *cf.* § 8.10 (Page 147). Les constantes propres à chaque routine sont initialisées ou, si possible, recalculées à l'aide des constantes fondamentales au premier appel de cette routine, elles y sont conservées par l'utilisation de l'instruction `SAVE`.

Des commentaires et références permettent de suivre et de vérifier les algorithmes. Dans chaque sous-programme les quantités entrantes et sortantes de la liste d'arguments sont répertoriées, parfois un peu succinctement. Dans un sous-programme, les variables et les routines externes au module auquel appartient le sous-programme sont introduites par l'instruction :

```
USE nom_module ONLY : routine, variable
```

Cette contrainte permet l'identification des MODULES d'appartenance des quantités invoquées.

Le nombre de couches, les limites zones radiatives / zones convectives variant en fonction du temps, il est fait grand usage de tableaux de dimensions variables qui sont alloués/déalloués en fonction des besoins.

Les notations utilisées pour les variables sont aussi explicites que possible, compte tenu d'une nécessaire concision *e.g.* on utilise `gradad` pour le gradient adiabatique $\nabla_{\rm ad}$; de même pour les dérivées *e.g.* les dérivées par rapport à $P_{\rm gaz} \equiv$ `p` ont un préfixe "d" et une extension[1] "p", exemple : dgradadp$\equiv \frac{\partial \nabla_{\rm ad}}{\partial P_{\rm gaz}}$.

8.3 Routine abon_ini

La routine abon_ini est une routine PRIVATE du module mod_nuc, *cf.* § E.9 (Page 261). Sa fonction principale est d'effectuer l'initialisation des abondances, des rapports isotopiques et des charges des éléments chimiques utilisés par le modèle. A la suite de cette initialisation dans la routine tabul_nuc, *cf.* Fig. 8.3 (Page 180), les éléments chimiques utilisés par les réactions thermonucléaires seront seuls retenus. La source des données est principalement le site d'OPAL : ftp : //www-phys.llnl.gov/pub/opal.

Description :
- Les charges, masses atomiques et les symboles des éléments chimiques jusqu'au nickel ($Z = 28$) sont initialisés puis un certain nombre de rapports isotopiques[2].
- Suivant la mixture initiale retenue, *cf.* § 3.15 (Page 33), les abondances en DeX (H=12) ou en nombre, sont initialisées puis, si besoin, transformées en abondances par nombre. Dans les cas où il y a utilisation d'une mixture initiale personnalisée, celle-ci est lue, en nombre, soit dans le fichier mon_modele.mix, soit dans le fichier mixture à défaut du précédent *cf.* § 4 (Page 37).
- Les abondances relatives des métaux dans Z sont ensuite évaluées.

Appel : abon_ini est appelée par la routine de calcul des réactions nucléaires lors de son initialisation, *cf.* § 8.76 (Page 179). abon_ini n'a **pas d'argument**

8.4 Routine add_ascii

Dans cette routine PRIVATE du module mod_cesam, *cf.* § E.13 (Page 264), on complète le tableau des quantités entrant éventuellement dans les fichiers ASCII de type output, *cf.* § 8.86 (Page 187).

Description :
En utilisant les valeurs du fichier des oscillations adiabatiques, il y a tabulation, en fonction du rayon, de X pour calcul de la dérivée $\frac{\partial X}{\partial r}$, ainsi que de $F_{\rm rad} = f(r)$ pour celui de $\frac{\partial F_{\rm rad}}{\partial r} = B - J$ et de $\int_0^r \frac{3}{4}\kappa\rho dr = f_{\rm edd}J(r)$. L'ordre des interpolations $m_{\rm x} \geq 2$ fixé dans la routine peut être adapté (ne pas dépasser 4 *i.e.* splines naturelles). Le fichier d'oscillations adiabatiques est ensuite augmenté, soit pour le calcul des inversions (ivar=25) soit pour celui des oscillations non adiabatiques (ivar=44). Les dérivées partielles du second ordre sont déduites numériquement en utilisant la routine df_rotx décrite au § 8.29 (Page 155). L'identification des quantités est décrite au § D.1 (Page 239).

Appel : add_ascii est appelée par cesam.

```
SUBROUTINE add_ascii(var,glob,itot,ivar)
```

[1]Cette règle est souvent transgressée pour raison de concision.
[2]Ces initialisations seraient plus justifiées dans la routine ctes.

- Entrées :
  - `glob` : Tableau des quantités globales.
  - `itot` : Nombre total de points.
  - `ivar` : Nombre de variables.
- Entrées/Sorties :
  - `var` : Tableau des variables.

8.5 Routine `alecian1`

La routine interne `alecian1` est une routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), sa fonction est le calcul des accélérations radiatives suivant le premier formalisme de G.Alécian. `alecian1` utilise les tables : `phi_psi2_100.dat ... phi_psi2_200.dat` du sous-directory `SUN_STAR_DATA`. Chaque fichier correspond à une valeur de la masse clairement identifiée. Cette routine est le regroupement et l'adaptation à CESAM2*k* des données calculées par les sous-programmes `modul_ppxi` et `modul_grad` de G.Alécian.

Description :

- Lors du premier appel : lecture des options, des données et préparation de tableaux nécessaires au calcul des accélérations. Les éléments chimiques utilisés sont identifiés et les isotopes sont regroupés. L'accélération radiative de chaque élément non identifié peut, soit être mise à 0, soit être fixée égale à $-g_\mathrm{e}$ (gravité), l'élément ne subit alors aucune force externe. Pour chaque élément identifié, bien que le calcul de l'accélération radiative puisse être effectué, l'une ou l'autre des possibilités décrites précédemment peut être éventuellement utilisée. Pour ce faire, il convient de suivre, ***avant compilation***, les intructions données en commentaires.

$\boxed{\textbf{PB}}$ `Dans la présente version de CESAM2k, les seuls éléments identifiables sont : C, N, O, Ne, Na, Mg, Al, Si, S, Ar, Ca, et Fe. Il n'est pas possible de calculer des accélérations radiatives pour H, He, Li, Be par exemple.`

- Puis, pour chaque appel : calcul des accélérations radiatives et calcul de la gravité effective sur chaque ion.

Appel : `alecian1` est appelée par la routine générique `f_rad`, *cf.* § 8.56 (Page 169).

```
SUBROUTINE alecian1(lum,ray,t,kap,dkapx,nel,ychim,ioni,grav,g_rad,dg_rad)
```

- Entrées :
  - `lum, ray, t, kap, dkapx` : luminosité, rayon, température, opacité, dérivée/X (mole).
  - `nel, ychim, ioni, grav` : nombre d'électrons libres par volume, composition chimique par mole, taux d'ionisation, gravité.
- Entrées/Sorties : `g_rad(i)` : vecteur des accélérations radiatives, la gravité est `grav+g_rad` sur l'élément d'indice i
- Sorties : `g_rad(i,j)` : matrice des dérivées des accélérations radiatives sur l'élément i / abondance par mole de l'élément j.

8.6 Routine ascii

Cette routine PRIVATE du module mod_cesam, *cf.* §E.13 (Page 264), permet la création d'un fichier de sortie ASCII personnalisé. Pour ce faire, coder nom_output='ascii' dans le fichier de données mon_modele.don. Les quantités à écrire, leur ordre et la structure souhaitée sont définis dans un fichier de nom sortie_ascii qui doit se trouver dans l'environnement, *cf.* §4.7 (Page 42). Les lignes d'identification sont adaptées aux Sorties effectuées. Le fichier ASCII ainsi créé aura l'extension -ascii, Exemple : mon_modele-ascii dont un exemple se trouve dans le sous-directory EXPLOIT.
Diverses possibilités sont offertes :
- Ecriture de la fraction de masse en réel ou en DeX *cf.* §D.1 (Page 239).
- Tabulation du centre vers la surface ou l'inverse.

Les variables et quantités globales à écrire sont **nécessairement** à prendre parmi celles des tableaux glob et var calculés par cesam *cf.* §B.1 (Page 233) et §B.2 (Page 233).

Description :
- Mise en correspondance des tableaux des quantités globales et des tableaux des variables créés par cesam et à écrire.
- Transformation éventuelle des masses en DeX ⇒ fraction de masse.
- Inversion éventuelle de l'ordre d'écriture des variables.
- Ecriture des en-têtes.
- Ecriture des variables.

Appel : Le sous-programme ascii est appelé par la routine générique output, *cf.* §8.86 (Page 187).

```
SUBROUTINE ascii(nglob,tglob,totvar,tvar,var,glob,itot,
1 logm,reverse,titre_ascii)
```

- Entrées :
  - nglob : Nombre de quantités globales à écrire.
  - tglob : Tableau des indices des quantités globales à écrire.
  - totvar : Nombre de variables à écrire.
  - tvar : Tableau des indices des variables à écrire.
  - var : Tableau des valeurs des variables créé par cesam.
  - glob : Tableau des valeurs des quantités globales créé par cesam.
  - itot : Nombre de points du tableau var.
  - logm=.TRUE. : Tabulation de la masse en DeX, en fraction de masse sinon.
  - reverse=.TRUE. : Tabulation du centre vers la surface, de la surface au centre sinon.
  - titre_ascii : Nom à attribuer au fichier ainsi créé.

8.7 Routine générique atm

Cette routine PUBLIC du module mod_atm, *cf.* §E.8 (Page 260), gère la restitution de l'atmosphère. Elle fait appel à différentes routines PRIVATE, suivant le type d'atmosphère requis défini par la variable nom_atm du fichier de données.

Dans ces routines y sont définies les conditions limites externes du problème différentiel de la structure interne ; ces conditions limites doivent porter sur deux des trois variables thermodynamiques P, T et ρ. Elles se trouvent localisées dans la région optiquement mince de l'atmosphère. Les équations de la structure interne supposent que le flux d'énergie radiative est transporté par diffusivité radiative *i.e.* approximation de diffusion, ce qui est inexact dans le milieu optiquement mince où sont définies les conditions limites externes. Il faut donc transporter ces limites du milieu mince dans le milieu épais. La source de CESAM2k offre la possibilité d'utiliser deux types de méthodes : l'approximation monocouche et la reconstitution de l'atmosphère. Se reporter, pour plus de détails, à Morel *et al.* [1994].

Appel : `atm` est appelée par les routines `cesam`, `static_m` et `static_r`.

```
SUBROUTINE atm(list,l_rac,r_rac,xchim,pt_rac,dptsdl,dptsdr,
1 t_rac,dtsdl,dtsdr,m_rac,dmsdl,dmsdr,p_rac,dpsdl,dpsdr,t_eff)
```

- Entrées :
  - `list=.TRUE.` : calcul réduit pour une liste (sans les dérivées),
  - `l_rac` : luminosité au raccord,
  - `r_rac` : rayon au raccord,
  - `xchim` : composition chimique par gramme.
- Sorties :
  - `pt_rac` : pression totale au raccord,
  - `dptsdl` : dérivée / L de la pression totale au raccord,
  - `dptsdr` : dérivée / R de la pression totale au raccord,
  - `t_rac` : température au raccord,
  - `dtsdl` : dérivée / L de la température au raccord,
  - `dtsdr` : dérivée / R de la température au raccord,
  - `m_rac` : masse au raccord,
  - `dmsdl` : dérivée / L de la masse au raccord,
  - `dmsdr` : dérivée / R de la masse au raccord,
  - `p_rac` : pression gazeuse au raccord,
  - `dpsdl` : dérivée / L de la pression gazeuse au raccord,
  - `dpsdr` : dérivée / R de la pression gazeuse au raccord,
  - `t_eff` : température effective.

8.8 Routine subordonnée `base_chim`

Cette routine subordonnée de `diffus`, génère le vecteur nodal de la base de B-splines utilisée pour la résolution des équations de la diffusion des éléments chimiques, *cf.* § 7.8 (Page 96).

Description :
- Allocation et initialisation du tableau des multiplicités.
- Définition des multiplicités aux limites zone radiative / zone convective.
- Formation du vecteur nodal.

Appel : Le sous-programme `base_chim`, n'a pas d'argument.

8.9 Routine subordonnée `base_rota`

Cette routine subordonnée de `diffus` et de `evol`, génère le vecteur nodal de la base de B-splines utilisée pour la résolution des équations de la diffusion du moment cinétique, *cf.* § 7.9 (Page 111).

Description :
- Allocation du tableau des multiplicités.
- Initialisation du tableau des multiplicités selon la présence ou l'abscence de limites zone radiative / zone convective.
- Formation du vecteur nodal.

Appel : Le sous-programme `base_rota`, n'a pas d'argument.

8.10 Routine `cesam`

Cette routine PUBLIC, sans argument, du module `mod_cesam`, *cf.* § E.13 (Page 264), constitue en fait le programme principal. Elle gère l'ensemble des calculs ; elle est appelée par les programmes `cesam2k` ou `cesam2k_dbg`, *cf.* § 8.11 (Page 148). Son organigramme général est présenté Figure 1.1 (Page 7). Elle est divisée en 5 parties :
- *Menu d'entrée identifiant l'option de* CESAM2k *à utiliser* :
  - Reprise d'une évolution en cours.
  - Initialisation d'un modèle de séquence principale d'âge zéro homogène.
  - Initialisation d'un modèle de pré-séquence principale homogène.

  Puis lecture du fichier de données `mon_modele.don`, enfin, initialisation des paramètres de calcul.
- *Cas de la reprise d'une évolution en cours.* Bien qu'il y ait séparation des cas, modèle repris en binaire ou en ASCII, le déroulement des calculs est assez similaire :
  - Lecture du modèle à reprendre, soit en binaire, soit en ASCII.
  - Vérification de la cohérence des paramètres utilisés dans le modèle repris et ceux définis par le fichier de données.
  - Initialisations diverses, en particulier les réactions thermonucléaires et détermination de la valeur du premier pas temporel à utiliser.
  - Pour un modèle repris en ASCII, initialisation des développements sur des bases de B-splines des variables principales et de la composition chimique.
- *Cas de l'initialisation d'un modèle de séquence principale d'âge zéro homogène.*
  - Séparation des cas : modèle d'initialisation en binaire, en ASCII.
  - Pour un modèle repris en ASCII, initialisation des développements sur les bases de B-splines des variables principales.
  - Formation de la fonction de répartition.
  - Initialisations diverses, en particulier les réactions thermonucléaires.
  - calcul du modèle de ZAMS homogène par appel à la routine `resout`.
- *Cas de l'initialisation d'un modèle de pré-séquence principale homogène.*
  - Séparation des cas : modèle repris en binaire ou en ASCII.
  - Pour un modèle repris en ASCII, initialisation des développements sur des bases de B-splines des variables principales.

  - Formation de la fonction de répartition.
  - Initialisations diverses, en particulier les réactions thermonucléaires.
  - Lecture du facteur de contraction, et calcul d'un premier modèle initial par appel à la routine `resout`.
  - Dialogue de validation du modèle calculé avec, éventuellement, reprise du calcul avec un facteur de contraction amélioré.
  - Calcul d'un second modèle initial avec un facteur de contraction augmenté de 10%.
  - Après validation, calcul du pas temporel initial.
- *Calcul de l'évolution.*
  - Ecriture du modèle en binaire pour les reprises : fichier `mon_modele_B.rep`. En affectant une valeur négative au paramètre `NB_MAX_MODELES` du fichier de données,*cf.* § 3.6 (Page 25), tous les fichiers binaires seront écrits dans l'environnement. Pour identification le numéro du modèle est indiqué dans le nom du fichier, exemple : `mon_modele`*0146*`_B.rep`.
  - Mise à jour du fichier `mon_modele.HR` pour dessin du diagramme HR ; ce dernier est réinitialisé dans le cas d'une nouvelle évolution.
  - Formation du listing `mon_modele.lis`.
  - Calcul du modèle au pas temporel suivant par appel à la routine `resout`.
  - Gestion du nouveau modèle :
    - Arrêt du calcul, formation du listing, du fichier ASCII s'il est demandé. On trouve au § B.1 (Page 233) et § B.2 (Page 233) la description des tableaux `glob` et `var` créés par `cesam` pour les Sorties ASCII.
    - Poursuite du calcul de l'évolution.

Appel : `cesam`, appelée par le programme principal, *cf.* § 8.11 (Page 148), ***n'a pas d'argument***.

8.11 Programmes `cesam2k, cesam2k_dbg`

Ces deux programmes permettent l'exploitation et le debug de CESAM2*k*. Ils ne comportent que l'instruction d'appel à la routine `cesam`, *cf.* § 8.10 (Page 147). Ils ne sont différenciés que par des options de compilation qui optimisent l'exécution ou permettent le debug. (Les deux programmes `cesamT` et `cesamT_dbg` identiques aux précédents et utilisés pour des tests, ne sont laissés dans la distribution que pour en faciliter la gestion.)

8.12 Routine `chim_gram`

La fonction de cette routine PUBLIC du module `mod_variables` effectue la transformation des valeurs des rapports d'abondances par mole et de leurs dérivées en rapports d'abondances par masse.

Appel : `chim_gram` est appelée de divers endroits, depuis `cesam` particulièrement.

```
SUBROUTINE chim_gram(xchim,dxchim)
```

- Entrées/Sorties :

– `xchim, dxchim` : composition chimique et dérivée par mole ⟶ par gramme.

8.13 Routine générique coeff_rota

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), est la gestion du calcul des coefficients de la diffusion du moment cinétique.

Description : Le calcul est orienté vers la routine `NOM_COEFF_ROTA` définie dans la routine `cesam.f`, *cf.* § 8.10 (Page 147), suivant la précision requise pour le calcul, *cf.* § 3.15 (Page 33).

Appel : `coeff_rota` est appelée par `eq_diff_rota`, *cf.* § 8.47 (Page 164).

```
SUBROUTINE coeff_rota(dt,nu,y,frl,ddfrl)
```

- Entrées :
  - `dt, nu, y` : pas temporel, abscisse, tableau des variables de la rotation.
- Sorties :
  - `frl, ddfrl` : coefficients des équations de la rotation et dérivées par rapport aux variables de la rotation.

8.14 Routine coeff_rota_ioni

Cette routine a pour fonction le calcul des coefficients des équations de la diffusion du moment angulaire. Le poids moléculaire moyen et les charges moyennes des isotopes étant obtenus en supposant tous les éléments chimiques ionisés.

Description :
- Interpolation au point de calcul des variables de l'équilibre quasi-statique.
- Estimation des variables de la rotation et de l'équilibre quasi-statique au pas temporel précédent.
- Interpolation au point de calcul des variables de la rotation.
- Calcul des coefficients non identiquement nuls.

Appel : `coeff_rota_ioni` appelé par `coeff_rota` a les mêmes arguments.

8.15 Routine coeff_rota_saha

Cette routine a pour fonction le calcul des coefficients des équations de la diffusion du moment angulaire. Le poids moléculaire moyen et les charges moyennes des isotopes étant obtenus en résolvant l'équation de Saha.

Description :
- Interpolation au point de calcul des variables de l'équilibre quasi-statique.
- Estimation des variables de la rotation et de l'équilibre quasi-statique au pas temporel précédent.
- Interpolation au point de calcul des variables de la rotation.
- Calcul des coefficients non identiquement nuls.

Appel : `coeff_rota_saha` appelé par `coeff_rota` a les mêmes arguments.

8.16 Routine `coeff_rota_z16`

Cette routine a pour fonction le calcul des coefficients des équations de la diffusion du moment angulaire. Le poids moléculaire moyen et les charges moyennes des isotopes sont obtenus en approchant la masse atomique moyenne et la charge moyenne des éléments lourds par celles de l'oxygène.

Description :
- Interpolation au point de calcul des variables de l'équilibre quasi-statique et calcul de divers coefficients.
- Interpolation de la composition chimique, appel à l'équation d'état.
- Estimation des variables de la rotation et de l'équilibre quasi-statique au pas temporel précédent.
- Calcul de l'opacité, des taux d'ionisation, de l'énergie nucléaire et calcul de divers coefficients.
- Interpolation au point de calcul des variables de la rotation.
- Calcul des coefficients de diffusion et des pertes de moment cinétique.
- Calcul des coefficients non identiquement nuls.

Appel : `coeff_rota_z16` appelé par `coeff_rota` a les mêmes arguments.

8.17 Routine `coll_atm`

La fonction de cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), est la résolution, dans l'espace des B-splines, du système des équations de la restitution de l'atmosphère Eq. 7.13 (Page 81).

Description :
- Initialisation du vecteur nodal et des limites, initialisations et allocations diverses lors du premier appel.
- Vérification, qu'au point de raccord, l'accélération centrifuge ne dépasse pas la gravité.
- Pour chaque point de collocation et chaque point limite, formation par appels à `eq_atm` des Eq. 7.13 (Page 81) dans l'espace des B-splines.
- Résolution du système et évaluation des corrections de la méthode itérative, retour à `lim_atm` ou poursuite des itérations (boucle infinie).

Appel : `coll_atm` est appelée par la routine `lim_atm` *cf.* § 8.67 (Page 173).

```
SUBROUTINE coll_atm(r_rac,l_rac,xchim,ord_atm,knot_atm,dim_atm,ms_atm)
```

- Entrées :
  - `r_rac, l_rac, xchim` : rayon, luminosité composition chimique au raccord avec l'enveloppe.
  - `ord_atm, knot_atm, dim_atm, ms_atm` : ordre des équations différentielles, nombre de points du vecteur nodal, dimension de la base de B-splines, ordre des B-splines.

8.18 Routine `coll_qs`

La fonction de cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), effectue la résolution, dans l'espace des B-splines, du sytème Eq. 7.5 (Page 71) des équations de la structure interne (équilibre quasi-statique).

Description :
- Initialisation du vecteur nodal et des limites, initialisations et allocations diverses lors du premier appel.
- Changement de la base de B-splines si le nombre de couches a varié depuis l'appel précédent.
- Pour chaque point de collocation et chaque point limite, formation par appels à `static`, *cf.* § 8.122 (Page 203), des équations 7.5 (Page 71) dans l'espace des B-splines.
- Résolution du système et évaluation des corrections et de la précision de la méthode itérative, retour à `resout` ou poursuite des itérations (boucle infinie).

Appel : `coll_qs` est appelée par `resout`, *cf.* § 8.114 (Page 195).

```
SUBROUTINE coll\_qs(dt,compt,reprend,err,vare,qmax,corr)
```

- Entrées :
  - `dt, compt, reprend=.TRUE.` : pas temporel, compteur du nombre d'itérations, il faut réinitialiser la solution.
- Sorties :
  - `err, vare, qmax, corr` : erreur maximale, variable présentant cette erreur, indice de la couche où l'erreur est maximale, correction.

8.19 Routine `collision`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), calcule les intégrales de collision $\Omega_{ij}^{(kl)}$, $k, l = 1, 2$ en utilisant les tabulations de Paquette *et al.* [1986]. Ces tables ont été scannées de l'ApJS. Les quantités qui figurent dans les `data` sont les valeurs des intégrales de collision calculées par les formules d'interpolation pour les $\psi_{st}(n)$ (notations de Paquette *et al.* [1986]), et prolongées jusqu'à $\psi_{st} = 4$; au delà de $\psi_{st} = 4$, on utilise les approximations analytiques.

Au premier appel de la routine, on calcule les coefficients d'interpolation (appel à `bsp1dn`, *cf.* § 9.2.2 (Page 215)) et on effectue quelques autres initialisations. De la longueur de Debye λ, on déduit les valeurs des ψ_{ij} d'où l'interpolation des intégrales de collision et le calcul de leurs dérivées par rapport à chacune des abondances ; distinction est faite entre potentiels attractifs et répulsifs. On économise des ressources de temps calcul en exploitant les symétries.

Appel : `collision` est appelée par la routine `diffm_br` *cf.* § 8.32 (Page 156).

```
SUBROUTINE collision(nb,zb,mij,xi,ro,drox,t,omega11,zij,zpij,zsij,
1 domega11,dzij,dzpij,dzsij)
```

- Entrées :

  - `nb, zb, mij, xi` : nombre de particules (1 : indice de ^{1}H, `nb` : indice pour les électrons), charges, masses réduites, abondances par mole.
  - `ro, drox, t` : densité, dérivée logarithmique par mole, température.
- Sorties :
  - `omega11, zij, zpij, zsij` : tables des intégrales de collision.
  - `domega11, dzij, dzpij, dzsij` : tables des dérivées des intégrales de collision.

8.20 Routine générique `conv`

Cette routine PUBLIC du module `mod_conv`, *cf.* § E.7 (Page 259), gére le calcul du gradient de température $\nabla \equiv \frac{\partial \ln T}{\partial \ln P}$ dans les zones convectives et de ses dérivées partielles par rapport aux variables locales. Suivant la formulation de la convection définie dans le fichier de données, *cf.* § 3.3 (Page 22), un appel ciblé dirige le calcul vers la routine *ad-hoc*.

Description :
La structure des routines de convection est la même pour toutes. Seules diffèrent les formulations *cf.* § 7.10.2 (Page 127). Des grandeurs spécifiques aux conditions physiques du calcul sont initialisées au premier appel. Puis la racine réelle de la cubique est recherchée en utilisant l'algorithme de Newton-Raphson. Enfin le gradient convectif, l'efficacité de la convection et leurs diverses dérivées sont calculés.

Appel : Le sous-programme `conv` est appelé par les routines `thermo`, *cf.* § 8.128 (Page 207), et `thermo_atm`, *cf.* § 8.129 (Page 208).

```
SUBROUTINE conv(r,krad,gravite,delta,cp,ro,hp,taur,gradrad,
1 gradad,der,gradconv,dgradkra,dgradgra,dgradel,dgradcp,dgradro,
2 dgradhp,dgradtaur,dgradgrad,dgradgad,
3 gam,dgamkra,dgamgra,dgamdel,dgamcp,dgamro,
4 dgamhp,dgamtaur,dgamgrad,dgamgad)
```

- Entrées :
  - `r, krad, gravite, delta, cp` : rayon, conductivité radiative, gravité, δ, $c_{\rm p}$.
  - `ro ,hp, taur, gradrad, gradad` : densité, échelle de hauteur de pression, épaisseur optique de la bulle convective, gradient radiatif, gradient adiabatique – éventuellement corrigé du terme $\frac{d \ln P_{\rm gaz}}{d \ln P}$, *cf.* § 7.10.3 (Page 128).
  - `der=.TRUE.` : le calcul des dérivées est requis.
- Sorties :
  - `gradconv, dgradkra, dgradgra, dgradel, dgradcp, dgradro, dgradhp, dgradtaur, dgradrad, dgradad` : gradient de température et dérivées.
  - `gam, dgamkra, dgamgra, dgamdel, dgamcp, dgamro, dgamhp, dgamtaur, dgamgrad, dgamgad` : efficacité de la convection et dérivées.

8.21 Routine conv_a0

En un point d'une zone convective, il semble irréaliste que la longeur de mélange puisse être supérieure à la distance qui sépare ce point de la plus proche limite de la zone convective. Ainsi qu'Eggleton [1972] l'a proposé, avec la routine PRIVATE `conv_a0` du module `mod_conv`, *cf.* § E.7 (Page 259), la longueur de mélange utilisée devient nulle à chaque limite zone radiative / zone convective. L'introduction de la distance exacte change la nature du problème qui devient intégro-différentiel ce qui complique considérablement la résolution numérique. Ce qu'évite la formulation locale d'Eggleton, *cf.* § 7.10.2 (Page 127).

8.22 Routine conv_cm

Dans la routine PRIVATE `conv_cm` du module `mod_conv`, *cf.* § E.7 (Page 259), le gradient de température dans une zone convective est calculé selon les prescriptions de Canuto & Mazitelli [1991] avec, pour longueur de mélange, $l = \alpha H_{\rm p}$, *cf.* § 7.10.2 (Page 127).

8.23 Routines conv_cgm_reza, conv_cm_reza

Dans la routine PRIVATE `conv_cm_reza` du module `mod_conv`, *cf.* § E.7 (Page 259), similaire à `conv_cm`, il est tenu compte de la quantité thermodynamique $\delta = (\frac{\partial \ln \rho}{\partial \ln T})_P$. Dans la routine `conv_cgm_reza`, le calcul du gradient convectif est effectué selon les prescriptions de Canuto *et al.* [1996] avec la prescription de Bernkopf (Heiter *et al.* [2002]), *cf.* § 7.10.2 (Page 127).

Ces deux routines ont été mises en œuvre et mises à la disposition des utilisateurs de CESAM2k par Reza Samadi, LESIA , Observatoire de Paris

8.24 Routine conv_jmj

Les spécificités de la routine de convection PRIVATE `conv_jmj` du module `mod_conv`, *cf.* § E.7 (Page 259), ont été décrites au § 7.10.2 (Page 127). La mise en œuvre de `conv_jmj` a bénéficié de la collaboration de J.Provost et de MJ.Goupil.

8.25 Routine coulomb

Cette routine PRIVATE du module `mod_evol` effectue le calcul du logarithme de Coulomb et de ses dérivées par rapport à la densité et à l'abondance en ^{1}H ou ^{4}He, suivant le formalisme simplifié de Michaud & Proffit [1993], p.250.

Appel : `coulomb` est appelée par la routine `diffm_mp`.

```
SUBROUTINE coulomb(zi,zj,thetae,ro,x,t,lnlambij,
1 lnlambijr,lnlambijx,cij,cijr,cijx)
```

– Entrées :

  - `zi, zj, thetae` : charges des éléments `i` et `j`, `thetae`=1 pour un mélange hélium/hydrogène ;
  - `ro, x, t` : densité, abondance par volume de ^{1}H ou ^{4}He, température.
- Sorties :
  - `lnlambij, lnlambijr, lnlambijx, cij, cijr, cijx` : logarithmes de Coulomb et dérivées par rapport à ρ et ^{1}H ou ^{4}He.

8.26 Routines `ctes_85`, `ctes_94`

Ces routines PRIVATE du module `mod_donnees` permettent l'initialisation de la plupart des éléments du module `mod_donnees`, *cf.* § E.3 (Page 248), qui regroupe les principales constantes physiques à utiliser.

- `ctes_85` : Routine de référence, dont la source principale des données de physique est le CRC Handbook of chemistry and physics 75th edition 1994, D.R. Lide *et al.* CRC press Boka Raton, Ann Arbor, London, Tokyo.
- `ctes_94` : Identique à `ctes_85` avec des valeurs plus récentes pour quelques constantes.

Appel : Suivant la variable `nom_ctes` du fichier de données *cf.* § 3.3 (Page 22), l'une ou l'autre est appelée par la routine générique `ini_ctes` *cf.* § 8.61 (Page 171), au début de l'exécution du sous-programme `cesam` *cf.* § 8.10 (Page 147). Ces routines **n'ont pas d'argument**.

8.27 Routine générique `des`

Cette routine PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), est la ***routine générique*** de dessin, elle utilise le logiciel PGPLOT *cf.* § A.2 (Page 222). Au fur et à mesure du calcul, le diagramme HR, la répartition en zone radiative / zone convective, les variables de structure et les abondances en fonction de la masse ou du rayon sont tracés.

Appel : `des` qui n'a que des arguments d'entrée est appelée par `cesam`.

```
SUBROUTINE des(fin,dt,teff)
```

- `fin=.TRUE.` : dernier appel, fermeture du dessin,
- `dt, teff` : pas temporel, température effective.

8.28 Routines `des_m`, `des_r`

L'une ou l'autre de ces routines PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), est appelée par la routine générique `des` suivant le type de dessin défini par la variable `nom_des` du fichier de données, *cf.* § 3.3 (Page 22). Le dessin est effectué respectivement en fonction de la masse (`nom_des='des_m'`) ou du rayon `nom_des='des_r'`), ou encore n'est pas effectué (`nom_des='no_des'`)

Description : Les algorithmes de ces deux routines sont similaires :

– Initialisations diverses relatives au logiciel PGPLOT.
– Tracé des abondances par unité de masse, ^{1}H et ^{4}He sont dessinées en valeurs réelles, les abondances des autres éléments sont normalisées sur [0,1], les coefficients de normalisation sont indiqués sur l'écran pour chaque variable.
– Tracé des variables de structure normalisées, pression, température, rayon ou masse et luminosité, les coefficients de normalisation sont indiqués sur l'écran pour chaque variable.
– Tracé du diagramme HR, sur les deux axes, les échelles sont déterminées lors du premier appel en fonction de la valeur de la masse initiale ; si le fichier `mon_modele.TeffL` existe dans l'environnement, tracé de la boîte d'erreur *cf.* § 4.9 (Page 45).
– Tracé des zones convectives, à l'issue de chaque pas temporel, un trait plein indique les intervalles de masse affectés par la convection.

Particularité de `des_r` :
Le rayon d'une étoile change au cours de l'évolution ; il arrive que les graphes de la représentation des variables de structure en fonction du rayon utilisent mal le domaine spatial de l'épure. Lorsque le rayon total devient, soit supérieur aux 9/10, soit inférieur aux 3/4, du domaine, on effectue un changement d'échelle qui a pour effet de ramener la surface aux 3/4 du domaine. Tant que l'échelle en rayon reste fixée, un tiret vertical – aux 3/4 du domaine – matérialise la valeur antérieure du rayon qui a servi à définir l'échelle. Cette disposition permet de suivre l'évolution du rayon en restant dans le cadre de l'épure.

8.29 Routine `df_rotx`

Cette routine PUBLIC du module `mod_etat`, *cf.* § E.5 (Page 259), constitue un opérateur qui permet de transformer les dérivées par rapport à P, T, X d'une fonction thermodynamique $f(P, T, X)$ en celles par rapport à ρ, T, X. En appliquant plusieurs fois l'opérateur on obtient les dérivées d'ordre supérieur.

Appel : `df_rotx` est appelée par `add_ascii`, *cf.* § 8.4 (Page 143).

```
SUBROUTINE df_rotx(p,ro,t,alfa,delta,dfdp_tx,dfdt_px,dfdx_pt,
1 drodx_pt,df_dro,df_dt,df_dx)
```

– Entrées :
  – `p, ro, t, alfa, delta` : pression, densité, température, $\frac{\partial \ln \rho}{\partial \ln P}$, $-\frac{\partial \ln \rho}{\partial \ln T}$,
  – `dfdp_tx, dfdt_px, dfdx_pt,drodx_pt` : $\frac{\partial f}{\partial P}$, $\frac{\partial f}{\partial T}$, $\frac{\partial f}{\partial X}$, $\frac{\partial \rho}{\partial X}$.
– Sorties :
  – `df_dro, df_dt, df_dx` : $\frac{\partial f}{\partial \rho}$, $\frac{\partial f}{\partial T}$, $\frac{\partial f}{\partial X}$

8.30 Fonction `dgrad`

Cette fonction PUBLIC du module `mod_static`, *cf.* § E.12 (Page 263), forme le critère de convection de Schwarzschild ou de Ledoux suivant les prescriptions décrites au § 7.10.1 (Page 126).

Description :

– Après quelques initialisations, appel à l'équation d'état et calcul de l'opacité
– Calcul du gradient radiatif avec le cas particulier du centre.
– Formation du critère de convection de Schwarzschild ou de Ledoux.

PB `Pour une raison inconnue, l'utilisation du critère de Ledoux est délicate.`

Appel : `dgrad` est appelée de `lim_zc`, *cf.* §8.69 (Page 174).

```
FUNCTION dgrad(pt,p,t,dlpp,xchim,m,l,r,dxchim,w)
```

– `pt, p, t, dlpp, xchim, dxchim` : Pression totale et gazeuse, température, $\frac{\partial \ln P_{\rm gaz}}{\partial \ln P}$, composition chimique et dérivée ;
– `m, l, r, w` : masse, luminosité, rayon, vitesse angulaire.

8.31 Routine générique `diffm`

Cette routine PRIVATE du module `mod_evol`, *cf.* §E.11 (Page 262), est la ***routine générique*** de calcul de la diffusion microscopique. Elle gère les descriptions `diffm_br`, `diffm_mp` et `diffm_0`, cette dernière, formelle, fixe à zéro les coefficients de diffusion microscopique.

Description :
Pour chaque élément du vecteur de composition chimique, l'équation de diffusion, avec les termes nucléaires, est formulée dans la routine `eq_dif_chim`. Les coefficients de diffusion microscopique sont calculés dans les routines de type `diffm`, et des coefficients de diffusion turbulente et de mélange (moment angulaire inclus) sont introduits par l'intermédiaire des routines de type `difft` *cf.* §8.34 (Page 157).

Appel : `diffm` est appelée par `eq_diff_chim`, *cf.* §8.46 (Page 163).

```
SUBROUTINE diffm(p,t,r,l,m,ro,drox,kap,dkapx,w,gradrad,dgradradx,xi,d,dd,v,dv)
```

– Entrées :
  – `p, t, r, lum, ltot, m` : pression, température, rayon, luminosité locale et totale, masse.
  – `ro, drox, kap, dkapx` : densité, dérivée/ X, opacité, dérivée / X,
  – `gradrad, dgradradx, xi` : gradient radiatif, dérivée / X. composition chimique par mole.
– Sorties :
  – `d, dd` : matrice des coefficients de diffusion microscopiques d_{ij} de $\frac{\partial x_j}{\partial m}$ et dérivées $\frac{\partial d_{ij}}{\partial x_k}$.
  – `v, dv` : vecteur des vitesses v_i des x_i et dérivées $\frac{\partial v_i}{\partial x_k}$.

8.32 Routine `diffm_br`

La routine PRIVATE `diffm_br` du module `mod_evol`, *cf.* §E.11 (Page 262), permet de calculer les coefficients de diffusion microscopique en résolvant les équations de transport selon le formalisme de Burgers, en tenant compte, éventuellement, des accélérations radiatives *cf.* § 8.5 (Page 144). La formulation du système des équations de Burgers utilisée dans CESAM2k, est en grande partie originale. Elle est décrite § 7.8.3 (Page 99).

Description :
- Au premier appel, initialisation de constantes, des masses réduites, allocation diverses, initialisation des accélérations radiatives.
- Calcul des taux d'ionisation, calcul du poids moléculaire moyen, de la gravité corrigée des accélérations radiatives et de leurs dérivées.
- Calcul des intégrales de collision, des coefficients de résistance Eq. 7.50 (Page 104) et de leurs dérivées.
- Calcul des coefficients $\bar{q_{ij}}$ Eq. 7.57 (Page 107), d_i Eq. 7.58 (Page 107), f_{ij} Eq. 7.59 (Page 107) et de leurs dérivées.
- Formation des matrices Δ Eq. 7.54 (Page 107), $\mathcal{M}$ Eq. 7.55 (Page 107), A_w Eq. 7.56 (Page 107), A_r Eq. 7.60 (Page 108), A_c Eq. 7.62 (Page 108), A_e Eq. 7.61 (Page 108).
- Calcul de ω par inversion du système linéaire, Eq. 7.63 (Page 108).
- Formation de la matrice des coefficients de diffusion Eq. 7.66 (Page 109), et du jacobien Eq. 7.68 (Page 109).
- Calcul du vecteur des vitesses de diffusion Eq. 7.64 (Page 108) et du jacobien Eq. 7.67 (Page 109).

8.33 Routine `diffm_mp`

La routine PRIVATE `diffm_mp` du module `mod_evol`, *cf.* § E.11 (Page 262), utilise la formulation simplifiée de Michaud & Proffit [1993] pour l'estimation des coefficients de diffusion microscopique. On suppose que les éléments lourds sont *éléments traces*, *i.e.* un élément autre que ^{1}H et ^{4}He ne diffuse que par rapport aux protons. Ce formalisme, plus simple mais moins général que celui de Burgers, **ne s'applique pas** dans un milieu privé de noyaux d'hygrogène, *i.e.* un cœur d'hélium.

Description :
- Au premier appel, initialisation de constantes, des masses réduites, allocations diverses.
- Calcul des logarithmes de Coulomb pour ^{1}H et ^{4}He.
- Calcul des coefficients de diffusion de ^{1}H et de ^{4}He et dérivées.
- Calcul des coefficients de diffusion pour les éléments tests par rapport à ^{1}H et dérivées.

8.34 Routine générique `difft`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), est la ***routine générique*** de calcul de la diffusion turbulente.

Appel : `difft` est appelée par `eq_diff_chim`, *cf.* § 8.46 (Page 163).

```
SUBROUTINE difft(melange,t,ro,drox,kap,dkapx,deff,gradad,gradrad,m,d,dd)
```

- Entrées :
  - `melange` : melange=.TRUE. on est dans une zone à mélanger,
  - `t, ro, drox, kap, dkapx, deff` : température, densité, opacité, et dérivées par rapport à l'abondance de ^{1}H **par gramme**, coefficient de diffusion turbulente due à la rotation,
  - `gradad, gradrad, m` : gradients adiabatique et radiatifs, abscisse de calcul.
- Sorties :
  - `d, dd, v, dv` : tableaux $d(i,j)$, $dd(i,j,k)$, $v(i)$, $dv(i,k)$, $i,j,k = 1,\ldots,n_{\rm elem}$.

8.35 Routine `difft_nu`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), effectue le calcul des coefficients de diffusion turbulente.

Description :
- Quelques initialisations sont effectuées au premier appel.
- En cas de mélange, la diagonale de la matrice de diffusion est affectée du coefficient de mélange convectif $d_M >> 1$ *cf.* § 7.10.6 (Page 130).
- Sans mélange, on ajoute aux termes diagonaux de la matrice de diffusion, les coefficients de diffusion turbulente `d_turb` *cf.* § 3.3 (Page 22), $d_{\rm eff}$ *cf.* § 7.9.5 (Page 114) et de viscosité radiative Re_ν, *cf.* Morel & Thévenin [2002].

Appel : `difft_nu`, est appelée par `difft` :

```
SUBROUTINE difft(melange,t,ro,drox,kap,dkapx,deff,d,dd)
```

- Entrées :
  - `melange` : melange=.TRUE. on est dans une zone à mélanger,
  - `t, ro, drox, kap, dkapx, deff` : température, densité, opacité, et dérivées par rapport à l'abondance de ^{1}H **par gramme**, coefficient de diffusion turbulente due à la rotation,
- Sorties :
  - `d, dd` : tableaux $d(i,j)$, $dd(i,j,k)$, $i,j,k = 1,\ldots,n_{\rm elem}$.

8.36 Routine `difft_gab`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), effectue le calcul des coefficients de diffusion turbulente. Elle exploite une suggestion de M.Gabriel : pour éviter la sédimentation de l'hélium est des éléments lourds dans la partie supérieure de l'enveloppe, il suffit de mélanger la partie de l'enveloppe jusqu'à la fin de la zone d'ionisation de l'hélium, localisée vers 10^6K.

Description :
- Quelques initialisations sont effectuées au premier appel.
- S'il y a mélange, ou si la tempétature est inférieure à 10^6K la diagonale de la matrice de diffusion est affectée du coefficient de mélange convectif $d_M >> 1$ *cf.* § 7.10.6 (Page 130).

– Sans mélange, on ajoute aux termes diagonaux de la matrice de diffusion, les coefficients de diffusion turbulente `d_turb` *cf.* § 3.3 (Page 22), $d_{\rm eff}$ *cf.* § 7.9.5 (Page 114).

```
SUBROUTINE difft_gab(melange,t,deff,d)
```

- Entrées :
  - `melange` : melange=.TRUE. on est dans une zone à mélanger,
  - `t,deff` : température, coefficient de diffusion turbulente due à la rotation,
- Sortie :
  - `d` : tableau $d(i,j)$.

8.37 Routine `difft_sun`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), calcule un coefficient de diffusion turbulente, sous la ZC externe solaire, suivant M. Gabriel [1997]. **Appel :** `difft_sun` est appelée par `eq_difft`,

```
SUBROUTINE difft_sun(melange,deff,gradad,gradrad,m,d)
```

- Entrées :
  - `melange` : melange=.TRUE. on est dans une zone à mélanger,
  - `deff, gradad, gradrad, m` : coefficient de diffusion turbulente due à la rotation, gradients adiabatique et radiatifs, abscisse de calcul.
- Sorties :
  - `d` : tableau $d(i,j)$ $i,j,\ldots,n_{\rm elem}$.

8.38 Routine `diffus`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), gère l'intégration numérique des équations de diffusion des espèces chimiques *cf.* § 7.8 (Page 96) et du moment cinétique *cf.* § 7.9 (Page 111).
Ces équations, du type intégro-différentiel, sont couplées. Pour des raisons d'instabilité numérique, il s'est avéré extrèmement délicat de les résoudre simultatément, force a été de les traiter séquentiellement, au prix d'une convervenge médiocre. Leur résolution numérique utilise la méthode des éléments finis de Galerkin, *cf.* § 7.7.1 (Page 93). Bien que la structure des algorithmes de résolution utilisés pour chacun des deux systèmes différentiels soient identiques deux routines distinctes ont été créées, pour conserver un maximum de souplesse.

Description :
- Au premier appel, initialisation et diverses écritures.
- Recherche du nombre et des indices des limites zone radiative / zone convective.
- Création du vecteur nodal pour la diffusion des éléments chimiques.
- Création du vecteur nodal pour la diffusion du moment cinétique.

– Résolution des équations de la diffusion du moment cinétique.
– Résolution des équations de la diffusion des éléments chimiques.
– Déallocations diverses.

Appel : `diffus` est appelée par `evol`, *cf.* § 8.55 (Page 167).

```
SUBROUTINE diffus(ok,dt,mc_tmp,nc_tmp)
```

– Entrées :
  – `dt` : pas temporel,
  – `mc_tmp,nc_tmp` : masses temporaires (abscisses langrangiennes), nombre de masses temporaires.
– Sorties :
  – `ok : ok=.TRUE.` les algorithmes de résolution de la diffusion du moment cinétique et des espèces chimiques ont convergé au niveau de précision requis.

8.39 Routine générique `diffw`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), est la ***routine générique*** de calcul des coefficients de diffusion turbulente horizontale, verticale et effective du moment cinétique, *cf.* § 7.9 (Page 111). Dans la présente version de CESAM2k elle ne gère que la description `diffw_mpz`.

Appel : `diffw` est appelée par `eq_diff_rota`, *cf.* § 8.47 (Page 164).

Arguments de `diffw`

```
SUBROUTINE diffw(dlnro,grand_k,nu,n2mu,n2t,r,ro,dfrot,frot,
1 deff,dh,dv)
```

– Entrées :
  – `dlnro,grand_k,nu,n2mu,n2t` : gradient $\ln\rho$, K, ν, variables liées à la fréquence de Brunt-Väissälä, *cf.* § 7.9.5 (Page 114).
  – `r,ro,dfrot,frot` : rayon, densité, vecteur des variables de la rotation et dérivées.
– Sorties :
  – `deff,dh,dv` : coefficients de diffusion turbulente effective, horizontale, verticale.

8.40 Routine `diffw_mpz`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), effectue le calcul des coefficients de diffusion horizontale, verticale et effective du moment cinétique, selon les prescriptions de Mathis *et al.* [2004].

Description de `diffw_mpz` :
– Au premier appel, initialisation et diverses écritures.
– Calcul de $D_{\rm h}$
– Calcul de $D_{\rm v}$
– Calcul de $D_{\rm eff}$

Appel : `diffw_mpz`, appelée par `diffw`, utilise la même liste d'appel.

8.41 Routine dnunl

La fonction de cette routine PRIVATE du module mod_cesam, *cf.* § E.13 (Page 264), est le calcul approximatif, valable uniquement pour le cas solaire, de ν_0, $\delta\nu_{02}$ et $\delta\nu_{13}$ suivant Schatzman & Praderie [1990], formules 100, 101 et 102 p. 389.

Description de dnunl :
- Tabulation de la vitesse du son en fonction de R^2, calcul de ν_0 par intégration numérique de type Gauss (on utilise le fait que $rdr = 1/2dr^2$).
- Calcul de $\delta\nu_{02}$ et $\delta\nu_{13}$.

Appel : dnunl est appelée de list, *cf.* § 8.70 (Page 176).

```
SUBROUTINE dnunl(r,c,n,rtot,nu0,dnu02,dnu13,a)
```

- Entrées :
  - r, c, n, rtot : tables des rayons, des vitesses du son, nombre de points, rayon total.
- Sorties :
  - nu0, dnu02, dnu13, a : ν_0, $\delta\nu_{02}$, $\delta\nu_{13}$, le A de la formule 100.

8.42 Routine subordonnée ecrit_ascii

Cette routine subordonnée de la routine cesam a pour fonctions la formation la mise en forme et l'écriture des fichiers de sortie ASCII, *cf.* § 3.4 (Page 24). Pour décrire finement le comportement de la fréquence de Brunt-Väissälä, dans les fichiers de sortie destinés aux exploitations sismologiques, *cf.* § 7.10.7 (Page 130), des points sont ajoutés au voisinage des limites zones radiatives / zones convectives ; typiquement, avec la valeur du paramètre $l_0 = 3$, 20 points supplémentaires sont ajoutés de part et d'autre de chaque limite.

Description :
- Au premier appel, allocation du tableau des quantités globales.
- Allocations de tableaux et création des abscisses supplémentaires permettant de décrire finement le profil de la fréquence de Brunt-Väissälä au voisinage des limites zones radiatives / zones convectives. Par défaut, on dispose 10 abscisses supplémentaires de part et d'autre de chaque limite. Insertion de ces abscisses dans le tableau des abscisses connues.
- Formation de la table des quantités globales.
- Allocation du tableau des quantités variables et formation de la partie atmosphère.
- Formation de la partie structure interne et composition chimique. Les quantités correspondants aux abscisses déjà connues sont reprises ; celles correspondant aux point ajoutés sont calculées.
- Déallocations de tableaux.
- Appel à la routine add_ascii, *cf.* § 8.4 (Page 143) pour complément de données.
- Appel à la routine générique output, *cf.* § 8.86 (Page 187), qui oriente vers le type de sortie ASCII désirée.

– Déallocation du tableau des variables.

Appel : `écrit_ascii`, qui n'a pas d'argument, est appelée de `cesam` à l'occasion de la sortie de fin de calcul, ou encore à l'issue de chaque pas temporel si on désire conserver tous les fichiers ASCII, *cf.* § 3.4 (Page 24).

8.43 Routine subordonnée `ecrit_rota`

Cette routine subordonnée de `diffus` et de `evol`, a pour fonction l'écriture, dans l'évironnement du calcul, du fichier des variables et des coefficients des équations de la diffusion du moment cinétique, *cf.* § 7.9 (Page 111). Ce fichier est destiné à être exploité par le programme de dessin `des_rota.f` du sous-directory `EXPLOIT`, *cf.* § 3.17 (Page 35). Ce fichier de travail n'est pas destiné à des exploitations mais aux mises au point. Il n'est créé que si la variable logique `ecrit_rot`, *cf.* § 4.8 (Page 43), du module `mod_donnees` est vraie, il est alors formé à l'issue de chaque appel à la routine `resout_rota`, *cf.* § 8.116 (Page 198).

Description :
- Définition du nombre de variables et allocation du tableau à écrire.
- Calcul des coefficients des équations de la diffusion du moment cinétique, aux abscisses du vecteur de la rotation (le centre est omis en raison de la singularité de l'échelle de hauteur de presion).
- Formation du tableau des variables et écriture sur le fichier `mon_modele_coeff_rota.dat`. Ce dernier est écrasé par chaque nouvelle écriture.

Appel : Le sous-programme `ecrit_rota`, n'a pas d'argument.

8.44 Routine `edding`

Cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), calcule la température dans une atmosphère purement radiative, pour une profondeur optique grise donnée suivant la loi $T(\tau)$ d'Eddington.

Appel : `edding`, appelée par la routine générique `tdetau` *cf.* § 8.127 (Page 207), utilise les mêmes arguments.

8.45 Routine `eq_atm`

La fonction de cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), est de former les équations de restitution de l'atmosphère Eq. 7.13 (Page 81).

Description de `eq_atm` :
- Au premier appel, après quelques initialisations, on projette sur la base de B-spline l'atmosphère solaire, connue sous forme de DATA, on obtient ainsi la solution initiale, on définit ensuite les limites et on détermine les points de collocation.
- Extraction des grandeurs physiques, calcul de la $T_{\rm eff}$, de la gravité et loi $T(\tau)$.

- Calcul des grandeurs thermodynamiques, de ∇ en particulier, et dérivées, formation des équations aux points de collocation et aux points limites.

Appel : eq_atm est appelée de coll_atm, *cf.* § 8.17 (Page 150) :

```
SUBROUTINE eq_atm(fait,li,xchim,cx,y,be,ae,r_rac,l_rac,xcoll_atm)
```

- Entrées :
  - fait, li, xchim, cx : fait=1 point de collocation, fait=2 point limite, indice de la limite, composition chimique/gramme, indice du point de collocation,
  - y : variables et dérivées,
  - r_rac,l_rac, xcoll_atm : rayon au fond de l'atmosphère, luminosité, abscisses des points de collocation.
- Sorties :
  - be, ae : seconds membres et jacobien.

8.46 Routine eq_diff_chim

La routine PRIVATE eq_diff_chim du module mod_evol, *cf.* § E.11 (Page 262), a pour fonction de calculer, dans l'espace physique, les résidus des équations 7.40 (Page 98), les quantités figurant à gauche des produits scalaires et leurs dérivées par rapport aux abondances.
Pour un élément chimique d'indice i fixé, les coefficients de l'équation de diffusion calculés dans les routines de type diffm, *cf.* § 8.31 (Page 156) sont de trois sortes :

- Les coefficients v_i de x_i, dans l'équation 7.33 (Page 96), codés v(i).
- Les coefficients de diffusion microscopique $d^\star_{i,j}$, de $\frac{\partial x_j}{\partial m}$, dans l'équation 7.34 (Page 97), codés d(i,j).
- Les coefficients $d_{i,i}$, de $\frac{\partial x_j}{\partial m}$, dans l'équation 7.34 (Page 97), sont ajoutés à d(i,i) dans les routines de type difft, *cf.* § 8.34 (Page 157).

les dérivées par rapport à $x_k,\ k = 1, n_{\rm elem}$ sont respectivement notées dv(i,k) = dv(nbelem(k-1)+i) et dd(i,j,k) = dd(nbelem(nbelem(k-1)+j-1)+i).

Description de eq_diff_chim

- Au point courant :
  - Initialisations diverses,
  - extraction des variables de structure au point de calcul cx, appel à l'équation d'état, détermination des opacités, calcul des réactions thermonucléaires et des coefficients de l'équation de diffusion,
  - détermination des résidus des équations 7.40 (Page 98), des quantités figurant à gauche des produits scalaires et de leurs dérivées par rapport aux abondances.
- Au point limite externe, évaluation et exploitation de la perte de moment angulaire.

Appel : eq_dif_chim est appelée par resout_chim, *cf.* § 8.115 (Page 198).

```
SUBROUTINE eq_diff_chim(nu,y,dt,ad,as,bd,bs)
```

- Entrées :
  - `nu, dt, y` : point de calcul en $m^{\frac{2}{3}}$, pas temporel, variables et dérivées.
- Sorties :
  - `as, ad, bs, bd` : dérivées par rapport aux abondances $x_{i,j}$ des arguments des produits scalaires avec $S_j^{m_c}$ et $\frac{\partial x_{i,j}}{\partial \nu}$ et résidus.

8.47 Routine eq_diff_rota

La routine PRIVATE `eq_diff_rota` du module `mod_evol`, *cf.* § E.11 (Page 262), a pour fonction de calculer, dans l'espace physique, les coefficients et les résidus des équations de la diffusion du moment cinétique, § 7.9 (Page 111).

Description de `eq_diff_rota` :
- Interpolation au point d'abscisse lagrangienne `nu` des coefficients des équations.
- Détermination de la nature du milieu convectif ou radiatif par interpolation du vecteur de convection.
- Au point courant (`fait=1`) :
  - Formation des équations dans le milieu convectif.
  - Formation des équations dans le milieu radiatif.
- Sur les limites (`fait=2, 3`) :
  - Partie intégrée des équations pour un noyau convectif et une zone convective externe.

Appel : `eq_rota` est appelée par `resout_rota`, *cf.* § 8.116 (Page 198).

```
SUBROUTINE eq_diff_rota(fait,nu,y,ad,as,bd,bs)
```

- Entrées :
  - `fait, nu, y` : fait=1 : point courant, fait=2, 3 : limite, point de calcul en $m^{\frac{2}{3}}$, variables et dérivées.
- Sorties :
  - `as, ad, bs, bd` : dérivées par rapport aux variables Ω, U, θ, Λ, ψ des arguments des produits scalaires avec $S_j^{m_c}$ et $\frac{\partial x_{i,j}}{\partial \nu}$, résidus.

8.48 Routine générique etat

Les routines de type `etat` constituent les interfaces entre CESAM2k et la routine de calcul de l'équation d'état qui a souvent une origine externe. Ces routines sont des routines PRIVATE du module `mod_etat`, *cf.* § E.5 (Page 259).
Pour une pression gazeuse P, une température T et une composition chimique X_i données, `etat` calcule la densité ρ, l'énergie interne spécifique U, $c_{\rm p}$, δ, $\nabla_{\rm ad}$, et leurs dérivées premières, α, β, Γ_1. Certaines routines d'équation d'état utilisent des tables que l'on doit créer dans le sous-directory `SUN_STAR_DATA`, *cf.* § 2.2 (Page 12). Le nom de ces tables doit être indiqué dans le fichier de données. La ***routine générique*** de calcul de l'équation d'état est `etat`, routine PUBLIC du module `mod_etat`.

Appel : `etat` est appelée de divers endroits, en particulier depuis `evol`, `thermo`, `thermo_atm`, `eq_dif_chim`, `cesam`.

```
SUBROUTINE etat(p,t,xchim,deriv,
1 ro,drop,drot,drox,u,dup,dut,dux,
2 delta,deltap,deltat,deltax,cp,dcpp,dcpt,dcpx,
3 gradad,dgradadp,dgradadt,dgradadx,alfa,beta,gamma1)
```

- Entrées :
  - `p, t, xchim, deriv` : pression, température, composition chimique par gramme, `deriv=.TRUE.` calcul des dérivées ;
- Sorties :
  - `ro, drop, drot, drox` : densité et dérivées ;
  - `u, dup, dut, dux` : énergie interne et dérivées ;
  - `delta, deltap, deltat, deltax` : δ et dérivées ;
  - `cp,dcpp,dcpt,dcpx` : $c_{\rm p}$ et dérivées ;
  - `gradad, dgradadp, dgradadt, dgradadx` : $\nabla_{\rm ad}$ et dérivées ;
  - `alfa, beta, gamma1` : α, β, Γ_1.

8.49 Routine `etat_ceff`

La routine PRIVATE `etat_ceff` du module `mod_etat`, *cf.* § E.5 (Page 259), utilise le formalisme CEFF. C'est une adaptation à CESAM2k des programmes originaux de J.Christensen-Dalsgaard par A.Baglin, M.Auvergne, Y. Lebreton, P.Morel & B.Pichon.
Les routines qui relèvent de ce package sont les suivantes :
`bilinc, clmnew, cvmgp, cvmgz, cvmgn, cvmgt, cvmgm, dmpeqs, derive, eoststst, eqstfc, eqstpc, eqstrh, eosder, f4der, f4n, f4mhd, hheion, hmnion, hvionac, inteffc, ismax, isamax, ismin, leq, neweta, omegac, phderc, prthvi, setgm1, setf4, seteqs, blockdata bleqstc, setcnsc, sdot, ssum, sscal, scopy, saxpy, storec, zeroc.`
Ces routines sont regroupées dans un même package et compilées simultanément avec `etat_ceff` ; elles sont donc des routines PRIVATE du module `mod_etat`, *cf.* § E.5 (Page 259). Diverses modernisations concernant la programmation ont été effectuées par B.Pichon et P.Morel. Le BLOCKDATA initial a été transformé en un tableau de paramètres. D'origine externe, ces routines ne sont pas détaillées.

PB `Cette routine qui demande beaucoup de ressources est peu robuste, son usage est délicat.`

8.50 Routine `etat_eff`

La routine PRIVATE `etat_eff` du module `mod_etat`, *cf.* § E.5 (Page 259), utilise le formalisme EFF, la source, rassemblée dans le package `EFF`, est une adaptation à CESAM2k des programmes originaux de J. Christensen-Dalsgaard par A. Baglin, M. Auvergne & P.Morel.
Les routines qui relèvent de ce package sont les suivantes :
`bilin, eqstf, eqstp, hviona, inteff, omega, phder, setcns, store, theffp, zero.`

Elles sont exploitées sous forme de routines subordonnées de la routine `etat_eff`. Le BLOCKDATA initial a été transformé en un tableau de paramètres.

8.51 Routine `etat_gong1`

Selon les prescriptions du "Solar Model Comparison Project" J. Christensen–Dalsgaard [1988], la routine PRIVATE `etat_gong1` du module `mod_etat`, *cf.* § E.5 (Page 259), suppose le milieu totalement ionisé, mais ne tient compte ni de la dégénérescence, ni de l'énergie interne d'ionisation, ni de la pression de radiation.

8.52 Routine `etat_gong2`

Suivant "Solar Model Comparison Project" J. Christensen–Dalsgaard [1988], la routine PRIVATE `etat_gong2` du module `mod_etat`, *cf.* § E.5 (Page 259), utilise le formalisme EFF simplifié, sans tenir compte, ni de la pression de radiation, ni de la dégénérescence. La mise en œuvre de `etat_gong2` a bénéficié de la collaboration de J. Provost.

8.53 Routine `etat_mhd`

La routine PRIVATE `etat_mhd` du module `mod_etat`, *cf.* § E.5 (Page 259), assure l'interface avec l'équation d'état solaire tabulée MHD de W.Däppen *cf.* Mihalas *et al.* [1988]. Les tables, en ASCII compressé, sont distribuées dans le package de l'équation d'état MHD, par W.Däppen. Les plus récentes *i.e.* avec "correction τ", sont disponibles à l'adresse FTP anonymous `usc.edu` directory `pub/astro-physics/mhd-oc/wd-evo` ; leur mise en place est décrite au § 2.2.5 (Page 18).
Les routines qui relèvent de ce package sont regroupées et compilées simultanément avec `etat_mhd` ; ce sont donc des routines PRIVATE du module `mod_etat`. Il s'agit de :
`fmttob, mhdpx, mhdpx1, mhdpx2, intpt, mhdst, mhdst1, rtab, rabu, quint, quintd, lir.`
Les noms des tables à utiliser sont indiqués directement dans `etat_mhd` il n'y a donc pas lieu d'en indiquer les noms dans le fichier de données. Ces tables doivent être créées par l'utilisateur, *cf.* § 2.2.5 (Page 18), et placées dans le sous-directory `SUN_STAR_DATA`, *cf.* § 2.2 (Page 12).

8.54 Routines `etat_opal`, `etat_opalX`, `etat_opalZ`

Ces routines PRIVATE du module `mod_etat`, *cf.* § E.5 (Page 259), assurent l'interface avec les diverses versions de l'équation d'état tabulée OPAL, Iglesias & Rogers [1991]. Les tables d'équation d'état, disponibles sur le site WEB `ftp ://www-phys.llnl.gov/pub/opal/eos/` sont très grandes.
Suivant les valeurs de l'abondance en métaux requise, il faut contruire les tables ainsi qu'il est décrit au § 2.2.4 (Page 15), et les placer dans le sous-directory

SUN_STAR_DATA, *cf.* § 2.2 (Page 12). Les routines utilisent les tables OPAL en binaire. La transformation des BLOCKDATA de la distribution sous la forme de paramètres, a permis des gains de place et de temps calcul appréciables. La mise en œuvre de ces routines a bénéficié de collaborations de S.Brun, J.M. Marques & L.Piau.
Bien qu'identiques, les routines utilisées pour la lecture et l'interpolation des tables utilisent des paramétrages différents. Il s'agit de :
`esac, gmass, quad, r_opal_bin, rad_sub ro,_new, t6rinterp`.
Elles sont introduites sous la forme de routines subordonnées et reproduites pour chacune des routines d'équation d'état OPAL.

8.55 Routine `evol`

La routine PUBLIC `evol` du module `mod_evol`, *cf.* § E.11 (Page 262), gère le mélange convectif, l'intégration temporelle de la composition chimique, éventuellement du moment cinétique. Un organigramme schématique est reproduit Figure 8.1 (Page 168).

Description

- Initialisations diverses, éliminations éventuelles pour le mélange convectif ***uniquement*** de limites zones radiatives / zones convectives trop externes, détermination si le modèle est totalement radiatif ou convectif.
- Détermination de la répartition en masse – en $(m/M_\odot)^{2/3}$ – utilisée pour l'interpolation de la composition chimique et du moment cinétique. Les points retenus sont, ou bien identiques aux points de la grille utilisée pour les variables globales, ou bien situés aux centres des intervalles entre deux points de cette grille, ou bien encore ceux de la *grille fixe*, *cf.* § 3.7 (Page 27). A défaut d'une description physique de l'évolution des discontinuités aux limites zones radiatives / zones convectives l'utilisation alternative des milieux de grille comme points d'interpolation crée une légère diffusion numérique destinée à stabiliser l'évolution. Aux voisinages du centre et des limites zones radiatives / zones convectives, des points supplémentaires sont ajoutés pour améliorer la précision. Cette dernière disposition n'est appliquée qu'avec l'utilisation d'une *grille fixe*.
- Les indices du début et de la fin de chaque zone convective sont ensuite identifiés dans la grille pour l'interpolation de la composition chimique ; en enlevant les `"c"` de commentaires, des algorithmes de test permettent des vérifications.
- Une alternative se présente alors : :
  - On tient compte de la diffusion microscopique avec ou sans diffusion du moment cinétique, il y a appel à la routine `diffus`, *cf.* § 8.38 (Page 159), sauf si le modèle est complètement convectif depuis le début de l'évolution – ou de la reprise du calcul. Au retour de la routine `diffus`, la précision du calcul de l'évolution des abondances est estimée.
  - Il y intégration et mélange convectif classiques, en tenant compte d'éventuels gains ou pertes de masse, et en estimant la précision du calcul de l'évolution des abondances :
    - d'abord pour les zones convectives où l'intégration de la composition chimique est réalisée de façon simultanée avec le mélange convectif,

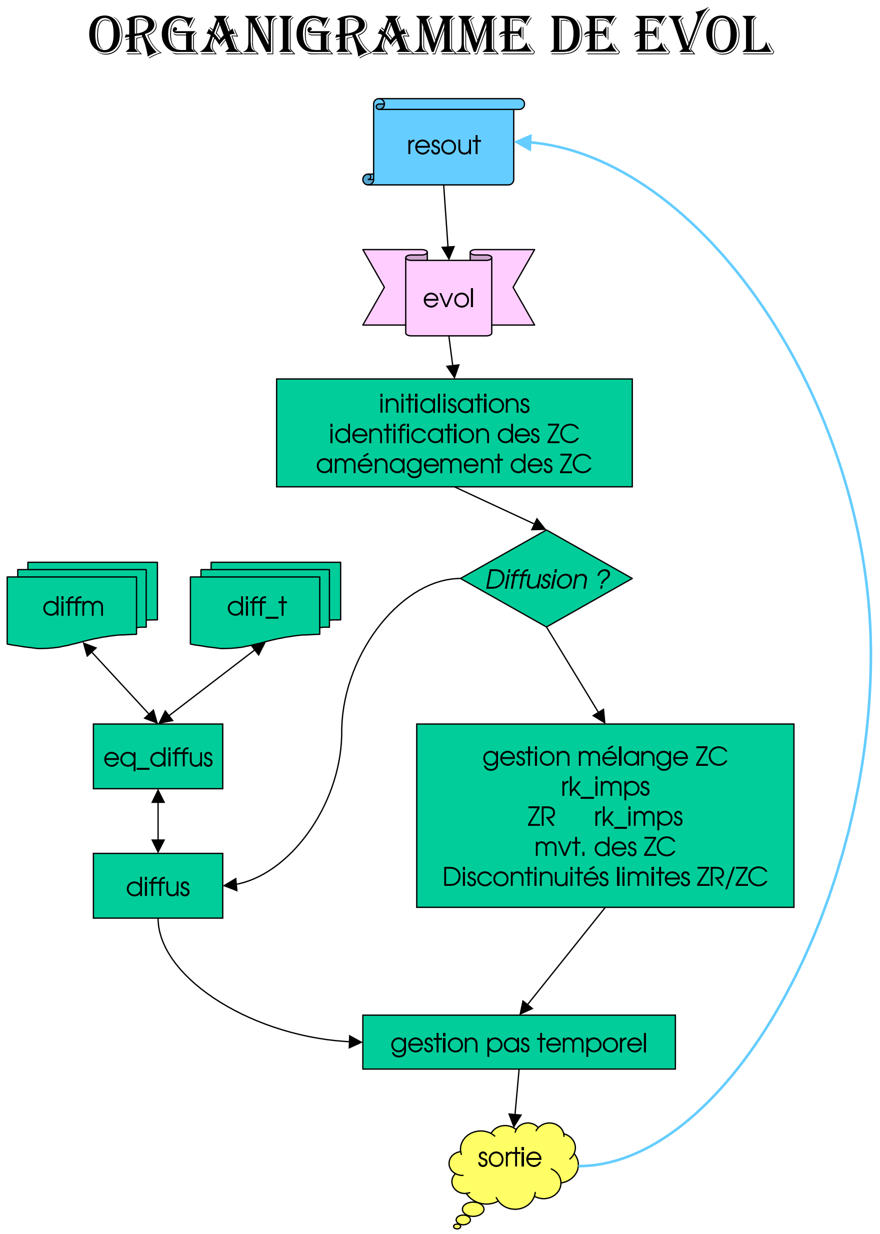

FIG. 8.1 – Environnement de la routine `evol` de gestion de l'évolution de la composition chimique et du moment cinétique.

  - ensuite pour les zones radiatives.
  - La composition chimique est ensuite ajustée en fonction des retraits ou avancées des limites zones radiatives / zones convectives.
  - Enfin, on calcule les coefficients d'interpolation de la composition chimique en tenant compte des discontinuités aux limites zones radiatives / zones convectives.
- Une renormalisation de la composition chimique de façon à assurer $X+Y+Z = 1$ est effectuée dans l'espace des splines. Cette renormalisation est nécessaire pour des arguments :
  - `Physiques :` les masses atomiques n'étant pas des nombres entiers, les équations de l'évolution thermonucléaire de la composition chimique ne peuvent conserver ***exactement*** $X + Y + Z = 1$. Pour ce faire, il faudrait tenir compte de corrections relativistes.
  - `Numériques :` les équations implicites de l'évolution de la composition chimique étant résolues de façon itérative, il y a nécessairement une erreur numérique résultant de l'appréciation de la convergence. Les erreurs d'arrondi sont aussi à prendre en ligne de compte.
- Une estimation du pas temporel à venir est faite en fonction des valeurs, requise et obtenue, de la précision de l'intégration. En cas de non convergence d'algorithmes itératifs, ou si la précision requise n'est pas atteinte, le modèle est réinitialisé lors du retour à `resout`, et le calcul est alors repris avec un pas temporel divisé par 2. ***L'estimation du pas temporel ne porte que sur l'estimation de la précision de l'intégration de la composition chimique.***

Appel : `evol` est appelée par `resout`, *cf.* §8.114 (Page 195).

```
SUBROUTINE evol(compt,dt,dts,reprend)
```

- Entrées :
  - `compt` : compteur des itérations globales, compt=0 pour la première itération globale de chaque nouveau pas temporel.
  - `dt` : pas temporel.
- Sorties :
  - `dts` : estimation du pas temporel à utiliser pour le pas temporel suivant.
  - `reprend=.TRUE.` : il faut réduire le pas temporel à cause de TdS ou de non convergence.

8.56 Routine générique `f_rad`

La routine PRIVATE `f_rad` du module `mod_evol`, *cf.* §E.11 (Page 262), constitue la ***routine générique*** du calcul des accélérations radiatives. Le calcul de l'accélération radiative est effectué avec la routine dont le nom `NOM_FRAD` est indiqué dans le fichier de données, *cf.* §3.9 (Page 29).

Appel : `f_rad` est appelée par `diffm_br`, *cf.* §8.32 (Page 156).

```
SUBROUTINE f_rad(lum,ray,t,kap,dkapx,nel,ychim,ioni,grav,g_rad,dg_rad)
```

- Entrées :
  - `lum, ray, t, kap, dkapx` : luminosité, rayon, température, opacité, dérivée/X (mole).
  - `nel, ychim, ioni, grav` : nombre d'électrons libres par volume, composition chimique par mole, taux d'ionisation, gravité.
- Entrées/Sorties : `g_rad(i)` : vecteur des accélérations radiatives, la gravité est grav+g_rad sur l'élément d'indice i
- Sorties : `dg_rad(i,j)` : matrice des dérivées des accélérations radiatives sur l'élément i par rapport à l'abondance par mole de l'élément j.

8.57 Routine subordonnée `fcmax`

Cette routine subordonnée de `resout` a pour fonction d'indiquer s'il sera nécessaire d'utiliser le nombre maximum de couches pour les modèles à calculer.

Description :
- Définition de la variable logique `logic` suivant la réalisation d'un des critères d'utilisation du nombre maximum de couches.
- Définition de la variable logique de sortie `cmax` suivant les critères de sortie ASCII définis au § D.1.4 (Page 243).

Appel : `fcmax` est appelé de `resout` à l'issue de la convergence du modèle.

```
SUBROUTINE fcmax(clogic)
```

- Sortie :
  - `clogic = .TRUE.` : il faudra utiliser le nombre maximum de couches.

8.58 Routine `from_alecian`

La subroutine PRIVATE `from_alecian` du module `mod_bp_for_alecian`, *cf.* § E.10 (Page 262), regroupe une partie des routines de calcul des accélérations radiatives, selon la seconde version du formalisme développé par G.Alécian. D'origine externe les fonctions des routines de ce regroupement ne sont pas détaillées.

8.59 Routine `hopf`

Cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), calcule la température dans une atmosphère purement radiative pour une profondeur optique grise donnée suivant la loi $T(\tau)$ de Hopf, *cf.* Mihalas [1978].

Description :
- Mise en forme des données pour l'interpolation.
- Interpolation de la température et dérivées.

Appel : `hopf`, appelée par la routine générique `tdetau` *cf.* § 8.127 (Page 207), utilise les mêmes arguments.

8.60 Routine iben

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), de type `nuc`, *cf.* § 8.76 (Page 179), calcule l'énergie gravifique pour dl'initialisation de l'évolution avec pré-séquence principale, *cf.* § 7.4.3 (Page 84). Cette initialisation est effectuée dans la routine `cesam`, *cf.* § 8.10 (Page 147) ; à cet effet, le nom de la routine de réactions thermonucléaires à utiliser est temporairement redéfini : `nom_nuc=iben`.

Appel : `iben`, appelée par la routine générique `nuc`. Les arguments sont ceux d'une routine de type `nuc`.

8.61 Routine générique ini_ctes

Cette routine PUBLIC du module `mod_donnees`, *cf.* § E.3 (Page 248), est la ***routine générique*** pour l'initialisation de la plupart des constantes physiques. Suivant le nom indiqué dans le fichier de données, *cf.* § 3.3 (Page 22), il est fait appel à la routine d'initialisation correspondante, *cf.* § 8.26 (Page 154).

Appel : Cette routine générique est appelée par `lit_nl`, *cf.* § 8.73 (Page 178), elle **n'a pas d'argument**.

8.62 Routine initialise_rota

La fonction de cette routine PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), est l'initialisation des fonctions décrivant la diffusion du moment cinétique.

Description :
- Détermination du rayon de l'étoile $R_\star$.
- Appels aux routines `w_initial`, *cf.* § 8.135 (Page 211), et `u_initial`, *cf.* § 8.132 (Page 210), pour l'initialisation de la vitesse angulaire Ω et de la vitesse de circulation méridienne U.
- Déduction des valeurs initiales de θ, Λ et ψ de celles de Ω et U.

Appel : `initialise_rota`, appelée par `cesam`, *cf.* § 8.10 (Page 147), n'a pas d'argument.

8.63 Routine inter

La fonction de cette routine PUBLIC du module `mod_variables`, *cf.* § E.4 (Page 255), est de déterminer par interpolation inverse, à partir de son développement sur une base de B-splines, l'indice de la couche correspondant à une masse ou à un rayon donné. Puis, par interpolation directe, de déterminer en ce point les valeurs des variables et de leurs dérivées premières, par rapport à la variable de masse ou de rayon. En variables lagrangiennes, l'interpolation est effectuée en $(m/M_\odot)^{2/3}$; en variables eulériennes l'interpolation est effectuée en $(r/R_\odot)$. La variable d'indice 6 *i.e.* $f(6)$, affectée à la fonction de répartition, *cf.* § 7.2.3 (Page 69), est utilisée pour transmettre l'indice décimal de la couche où s'effectue l'interpolation.

Appel : `inter` est appelée par divers programmes, en particulier `evol`.

```
SUBROUTINE inter(m23_ou_r2,bp,q,qt n,knot,x,f,dfdx,r2,m23)
```

- Entrées :
  - `m23_ou_r2` : chaîne de ***3 caractères***, si `m23_ou_r2='m23'` interpolation en masse, si `m23_ou_r2='r2␣'` interpolation en rayon.
  - `bp, q, qt, n, knot, x` : éléments pour l'interpolation.
  - `r2, m23` : points d'interpolation.
- Sorties :
  - `f, dfdx` : fonctions ($f(6)$: indice fractionnaire d'interpolation), dérivées.

8.64 Routine `inter_atm`

Pour l'atmosphère, cette routine PUBLIC du module `mod_exploit`, *cf.* § E.14 (Page 264), a une fonction similaire à la routine `inter` précédente. Elle est utilisée par le programme `for037_2k` du module `mod_exploit`, *cf.* § 3.17 (Page 35), qui permet l'interpolation d'un modèle en des masses ou des rayons définis à l'avance.

Description :
- Localisation de l'intervalle d'interpolation en masse, rayon ou indice de couche.
- Détermination de l'indice fractionnaire d'interpolation inverse par algorithme de Brent.
- Interpolation des variables de l'atmosphère.
- L'indice fractionnaire est conservé.

Appel : `inter_atm` est appelée par le programme `for037_2k`.

```
SUBROUTINE inter_atm(m_ou_r,ne_atm,bp_atm,x_atm,xt_atm,n_atm,
1 ord_atm,knot_atm,m_atm,r_atm,x,f,dfdx)
```

- Entrées :
  - `m23_ou_r2` : chaîne de 3 caractères, interpolation en masse si `m23_ou_r2='m23'`, ou interpolation en rayon si `m23_ou_r2='r2␣'`.
  - `ne_atm, bp_atm, x_atm, xt_atm, n_atm, ord_atm, knot_atm` : éléments pour l'interpolation par B-spline.
  - `m_atm, r_atm` : masses et rayons de la discrétisation de l'atmosphère.
- Sorties :
  - `f, dfdx` : fonctions, dérivées, rayon, masse.
  - `f(ne_atm+1)=x_int` : indice fractionnaire de l'abscisse d'interpolation.

8.65 Routines `k5750, k5777`

Ces deux routines PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), calculent la température dans une atmosphère non purement radiative pour une profondeur optique grise donnée, suivant les lois $T(\tau)$ solaires $T_{\rm eff} = 5750\,{\rm K}$ et $T_{\rm eff} = 5777\,{\rm K}$ calculées par C. Van't Veer avec le programme Atlas 9 de Kurucz.

Description :
- Mise en forme des données pour l'interpolation.
- Interpolation de la température et dérivées.

Appel : `k5750, k5777`, sont appelées par la routine générique `tdetau`, *cf.* § 8.127 (Page 207). Elles utilisent les mêmes arguments.

8.66 Routine `kappa_cond`

Cette routine PRIVATE du module `mod_opacite`, *cf.* § E.6 (Page 259), calcule l'opacité conductive en utilisant les ajustements polynômiaux de Iben [1975], et complète l'opacité radiative.

Description : Bien que remaniée pour son adaptation à CESAM$2k$, cette routine, en grande partie d'origine externe, n'est pas détaillée. La moyenne harmonique avec l'opacité radiative est effectuée à l'issue du calcul.

Appel : Cette routine est appelée par la plupart des routines de calcul d'opacité à l'issue du calcul de l'opacité radiative.

```
SUBROUTINE  kappa_cond(xh,t,ro,kappa,dkapdt,dkapdr,dkapdx)
```

- Entrées :
  - `xh` : composition chimique en fraction de masse.
  - `t` : température.
  - `ro` : densité.
- Entrées/Sorties :
  - `kap` : opacité.
  - `dkapdt, dkapdr, dkapdx` : dérivées de l'opacité en fonction de la température, densité, abondance d'hydrogène.

8.67 Routine `lim_atm`

Cette routine PUBLIC du module `mod_atm`, *cf.* § E.8 (Page 260), permet de restituer l'atmosphère à partir d'une loi $T(\tau)$ grise, *cf.* § 7.3.2 (Page 76).

Description :
- Au premier appel :
  - Initialisation de constantes.
  - Recherche, dans l'environnement, d'un fichier binaire d'atmosphère pour reprise, et tests de cohérence.
  - Reprise de l'atmosphère en binaire ou initialisation en ASCII.
  - Recherche du rayon bolométrique et initialisations diverses.
  - Projection sur la base de B-splines.
- Pour chaque appel :
  - Commentaires divers.
  - Appel à `coll_atm` pour l'intégration.
  - Au cas où une liste complète du modèle est requise, formation des tableaux des quantités nécessaires.
  - Ecriture du fichier binaire d'atmosphère `mon_modele.atm`.

Appel : `lim_atm` appelée par la routine générique d'atmosphère `atm`, *cf.* § 8.7 (Page 145), a la même liste d'appel.

8.68 Routines `lim_gong1, lim_tau1`

Ces deux routines PUBLIC du module `mod_atm`, *cf.* § E.8 (Page 260), sont deux formes similaires de la restitution d'une atmosphère dans l'approximation monocouche, *cf.* § 7.3.1 (Page 75). Dans la routine `lim_gong1`, deux facteurs $\beta = 7.22$ et $\lambda = 6$, affectent respectivement les équations de la pression et de la luminosité, leur effet étant, avec une équation d'état supposant l'ionisation totale, d'obtenir un modèle solaire qui ressemble au soleil actuel.

$\boxed{\textbf{PB}}$ `Avec l'approximation monocouche, la pression turbulente est ignorée dans l'atmosphère.`

Appel : `lim_gong1, lim_tau1`, appelées par la fonction générique d'atmosphère `atm`, *cf.* § 8.7 (Page 145), ont la même liste d'appel.

8.69 Routine `lim_zc`

Cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), a trois fonctions principales :

- Initialisation, si nécessaire, de la répartition des couches en fonction du nombre de points à utiliser.
- Localisation des limites des zones radiatives et convectives, au besoin avec overshoot.
- Détermination des coefficients de répartition.

`lim_zc` est une routine importante de CESAM2*k*, son organigramme est présenté Figure 8.2 (Page 175).

Description de `lim_zc`

- Initialisation :
  - Initialisations diverses effectuées au premier appel, en particulier, détermination des points limites et calcul des coefficients d'intégration correspondants à cette répartition.
  - Au cas où le nombre de couches doit être modifié, détermination de la nouvelle fonction d'espacement, changement de la base de B-splines, redéfinition des points limites et calcul des coefficients d'intégration. Avec l'option `grille_fixe`, *cf.* § 3.7 (Page 27), la grille d'interpolation de la composition chimique n'est modifiée que si le nombre de couches a varié de plus de la quantité `dn_fixe` fixée dans `resout`, *cf.* § 8.10 (Page 147).
- Localisation :
  - Calcul, en chaque point de la grille, de la différence $\nabla_{\rm rad} - \nabla^*_{\rm ad}$, localisation des changements de signe et identification de la nature de la limite à l'aide de la variable logique `lconv` qui est `.TRUE.` lorsqu'on passe d'une zone radiative à une zone convective, en traversant la limite dans le sens croissant de la variable d'espace *i.e.* masse, rayon ou encore, indice.
  - Elimination des zones convectives mal définies *i.e.* trop proches du centre, de la limite externe ou encore n'affectant que très peu de couches.
  - Affinement, par dichotomie, de la localisation des limites retenues.
  - Si besoin, extensions des zones convectives et analyse des chevauchements éventuels.

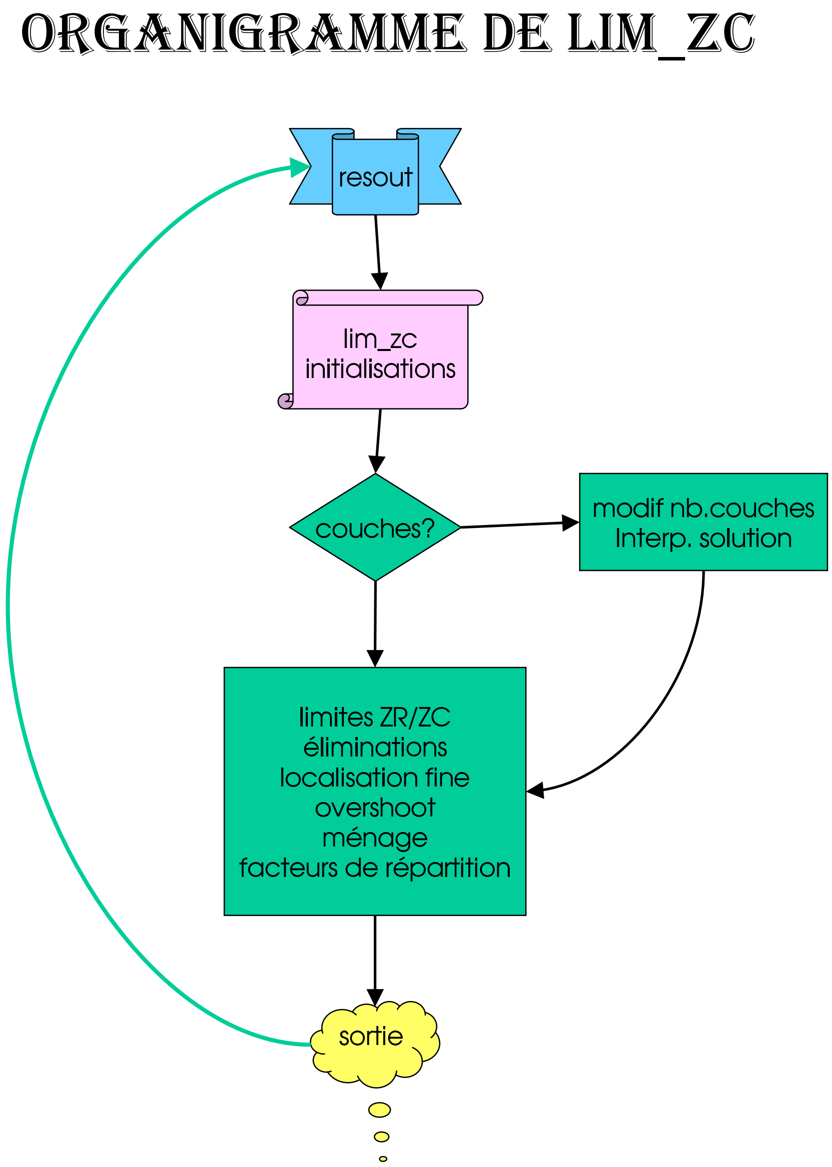

FIG. 8.2 – Organigramme de la routine `lim_zc` de détermination des limites zones radiatives / zones convectives.

- Détermination :
  - Des coefficients de répartition qui permettent d'amener chacune des limites sur un point de la grille.

Appel : `lim_zc`, appelée par `resout`, n'a que des arguments d'entrée.

```
SUBROUTINE lim_zc(no_rep,new_nqs)
```

`no_rep=.TRUE.` : Il n'y a pas eu d'initialisation ou de réinitialisation du modèle.
`new_nqs=.TRUE.` : Il faut effectuer une redistribution des couches du modèle quasi-statique, la constante de répartition ayant été modifiée.

8.70 Routine `list`

Cette routine PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), n'a que des arguments d'entrée. Sa fonction est de construire le listing permettant de suivre le déroulement du calcul et de produire, pour différentes époques de l'évolution, une liste détaillée des variables du modèle, des abondances, des rapports isotopiques à la surface et des flux de neutrinos – l'implantation de ces derniers a bénéficié d'une collaboration avec G.Berthomieu.

Description :
- Initialisations diverses, en particulier des indices d'identification des éléments chimiques.
- Construction d'un cartouche indiquant : l'âge du modèle, la température effective, la luminosité, le rayon, la gravité, la pression, la température, les abondances par fraction de masse de ^{1}H et ^{4}He centrales, les proportions d'énergie libérées par les cycles thermonucléaires et la gravité, la nature du modèle, la variation relative de masse, ses valeurs présente et initiale, les abondances et les rapports isotopiques en surface, éventuellement la période et la vitesse de rotation de surface.
- *Si le listing complet est requis*, le modèle est listé couche par couche du sommet de l'enveloppe vers le centre, un cartouche étant disposé toutes les 10 couches pour faciliter l'identification des quantités écrites. Les limites zones radiatives / zones convectives sont matérialisées. L'atmosphère est ensuite reproduite couche par couche, en partant de l'enveloppe vers la surface ; on indique ensuite la déplétion des éléments en surface, les abondances relatives en nombre à la surface, les flux de neutrinos, et une estimation, pour le cas solaire, des grandeurs astérosismologiques : ν_0, $\delta\nu_{02}$, $\delta\nu_{13}$, A.

Appel : `list` est appelée par `cesam`, *cf.* § 8.10 (Page 147).

```
SUBROUTINE list(alfa,anub8,anube7,anun13,anuo15,anupep,anupp,beta,
1 compg,cp,delta,dcapdr,dcapdt,depsdr,depsdt,d2p,d2ro,
2 chaine,convec,ecritout,epsilon,gamma,gamma_atm,gradad,grada_atm,
3 gradconv,gradc_atm,gradrad,gradr_atm,hp,i_cno,i_gr,i_pp,i_3a,
4 kap,l,m,mu,mue,m_atm,p,pt,pt_atm,p_atm,r,ro,ro_atm,r_atm,t,tau,
5 teff,tx_ioni,t_atm,u,vaissala,w,z,degene)
```

- `anub8, anube7, anun13, anuo15, anupep, anupp` : Tables des nombres de neutrinos des diverses sources.
- `alfa, beta, delta` : α, β, δ.
- `compg, cp, hp` : Composition chimique par unité de masse, chaleur spécifique à pression constante, échelle de hauteur de pression.
- `kap, dcapdr, dcapdt` : Opacité, dérivées par rapport à la densité et à la température.
- `epsilon, depsdr, depsdt` : Energie nucléaire des différents cycles et énergie graviphique, dérivées par rapport à la densité et à la température.
- `d2p, d2ro` : Dérivées secondes de la pression et de la densité centrales.
- `chaine, convec, ecritout` : Statut du modèle, nature radiative ou convective de la couche, écriture du listing complet.
- `gradad, gradconv, gradrad, gamma` : Gradient adiabatique, gradient convectif, gradient radiatif, Γ.
- `grada_atm, gradc_atm, gradr_atm, gamma_atm` : Gradient adiabatique, gradient convectif, gradient radiatif, Γ pour l'atmosphère.
- `i_cno, i_pp, i_3a, i_gr` : Pourcentages d'énergie d'origine nucléaire PP, CNO, 3α et gravifique.
- `mu, mue, vaissala` : Poids moléculaire moyen, poids moléculaire moyen par électron libre, fréquence de Brunt-Väissälä.
- `p, pt, t, r, l, m, ro, u` : Pression gazeuse, pression totale, température, rayon, luminosité, masse, densité, énergie interne spécifique.
- `p_atm, pt_atm, t_atm, r_atm, m_atm, ro_atm` : Pression gazeuse, pression totale, température, rayon, masse, densité dans l'atmosphère.
- `tau, teff` : Epaisseur optique, température effective
- `tx_ioni, w, z, degene` : Taux d'ionisation, vitesse angulaire, abondance des métaux, facteur de dégénérescence.

8.71 Routine `lit_binaire`

Cette routine PUBLIC du module `mod_exploit`, *cf.* § E.14 (Page 264), permet la lecture des fichiers binaires, *cf.* § C (Page 237), à l'exception des fichiers d'atmosphère.

Description :
- Recherche et lecture du fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22).
- Lecture des paramètres du modèle, calcul des dimensions et allocation des tableaux.
- Lecture du modèle.

Appel : `lit_binaire` appelée par des programmes d'exploitation, *cf.* § 3.17 (Page 35), du module `mod_exploit`, *cf.* § E.14 (Page 264), n'a que des arguments d'entrée.

```
SUBROUTINE lit_binaire(chaine,dt)
```

- `chaine, dt` : Nom du modèle, pas temporel.

8.72 Routine `lit_hr`

Cette routine PUBLIC du module `mod_exploit`, *cf.* § E.14 (Page 264), permet la lecture des fichiers ASCII de diagramme HR `mon_modele.HR`, *cf.* § D.2 (Page 244).

Description :
- Lecture du premier enregistrement du fichier ASCII `mon_modele.HR`, pour allocations.
- Lecture du fichier dans son intégralité.
- Formation des tests de lecture.

Appel : `lit_hr` est appelée par des programmes d'exploitation, *cf.* § 3.17 (Page 35), du module `mod_exploit`, *cf.* § E.14 (Page 264), en particulier par le programme `des_hr`.

```
SUBROUTINE lit_hr(init,chaine,fin,erreur,log_l,log_r,log_teff,
1 vrot,wrot)
```

- Entrées :
  - `init=.TRUE., chaine` : Initialisation, nom du modèle.
- Sorties :
  - `fin=.TRUE., erreur=.TRUE.` : Fin de fichier, erreur de lecture.
  - `log_l, log_r, log_teff` : Logarithmes décimaux de la luminosité, du rayon, de la température effective.
  - `vrot, wrot` : vitesses linéaire et angulaire de la couche externe.

8.73 Routine `lit_nl`

La fonction de cette routine PUBLIC des modules `mod_donnees`, *cf.* § E.3 (Page 248) et `mod_exploit`, *cf.* § E.14 (Page 264), est de lire les NAMELISTS du fichier d'entrée `mon_modele.don`, *cf.* § 3.3 (Page 22).

Description :
- Recherche et lecture des NAMELISTs du fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22). En cas d'échec de lecture, tentative de trouver et d'utiliser des fichiers de données de versions précédentes de CESAM.
- Vérification de la cohérence des données.
- Détermination des abondance par masse de l'hydrogène X, hélium Y et métaux Z, mise en place de tests.
- Initialisation des principales constantes de physique par appel à la routine `ini_ctes`, *cf.* § 8.61 (Page 171).
- Définition du type de rotation à utiliser.

Appel : `lit_nl` est appelée par `cesam`, *cf.* § 8.10 (Page 147).

```
SUBROUTINE lit_nl(wrot)
```

- Sortie :
  - `wrot` : Vitesse angulaire.

8.74 Fonction logique `lmix`

La valeur de cette fonction PRIVATE du module `mod_evol` est .TRUE. dans les zones convectives.

Description :
- Un premier test détermine si on se trouve dans la zone convective externe.
- Une fois la localisation de l'abscisse obtenue, on affecte à la fonction la valeur de la table de mélange pour l'indice obtenu.

Appel : `lmix` est appelée principalement des routines concernées par la diffusion du moment cinétique et des éléments chimiques.

```
LOGICAL FUNCTION lmix(nu)
```

- `nu` : $(\text{masse})^{\frac{2}{3}}$.

8.75 Routine `modif_mix`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), permet de personnaliser une mixture, *cf.* § 4.2 (Page 38).

Description :
- Recherche du fichier de la nouvelle mixture `mon_modele.modif_mix`, *cf.* § 4.2 (Page 38).
- Lecture des modifications en DeX des abondances initiales.
- Modification des abondances des éléments lourds.

Appel : `modif_mix`, appelée par `abon_ini`, *cf.* § 8.3 (Page 143), n'a pas d'argument.

8.76 Routine générique `nuc`

La routine PUBLIC `nuc` du module `mod_nuc`, *cf.* § E.9 (Page 261) est la ***routine générique*** qui, par l'intermédiaire de la routine spécifique dont le nom est défini dans le fichier de données *cf.* § 3.3 (Page 22), gère le calcul des réactions et de l'énergie nucléaire, *cf.* § 7.11 (Page 134). Les routines de type `nuc` sont disponibles dans 4 catégories différentes, suivant le domaine de température couvert :
- ***test*** : réseaux de réactions simplifiés destinés à des tests.
- ***pré-3***α : utilisables depuis la pré-séquence principale jusqu'à l'amorçage des réactions de l'hélium, intervalle de température couvert : 0.5-100 MK. Les réseaux de réactions utilisés décrivent les cycles PP et CNO.
- ***3***α : utilisables depuis la pré-séquence principale jusqu'à l'amorçage des réactions du carbone, intervalle de température couvert : 0.5-200 MK. Les réseaux de réactions utilisés décrivent les cycles PP, CNO et 3α.
- ***CO*** : utilisables depuis la pré-séquence principale jusqu'au cycle de l'oxygène, intervalle de température couvert : 0.5-400 MK. Les réseaux de réactions utilisés décrivent les cycles PP, CNO, 3α, la combustion du carbone et de l'oxygène.

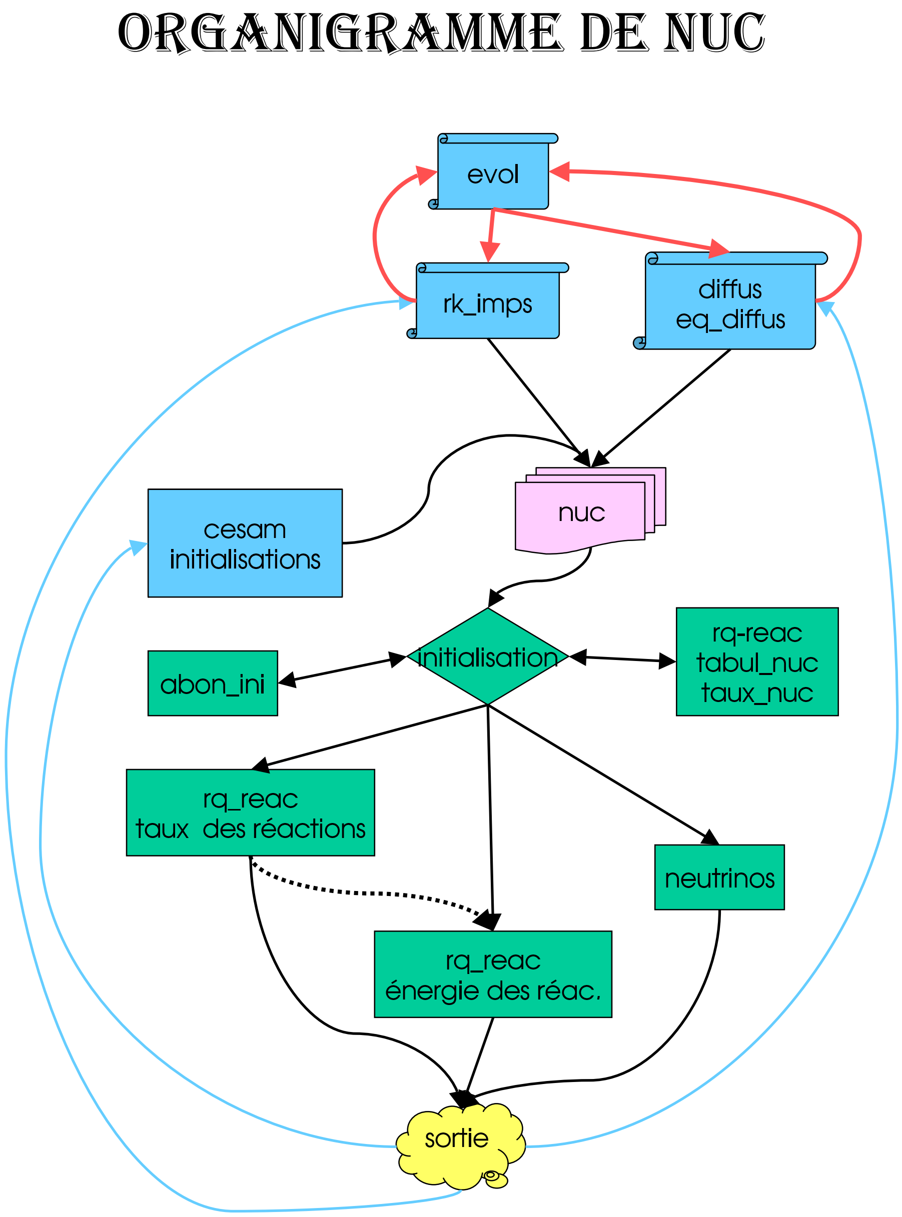

Fig. 8.3 – Environnement de la routine de calcul des réactions thermonucléaires. La routine `nuc` est appelée de `cesam` pour l'initialisation des abondances. Lors d'une évolution, elle est appelée i) de `eq_dif_chim` ou de `rk_imps` pour le calcul de l'évolution des abondances, suivant qu'il y ait ou non suivi de la diffusion des éléments chimiques, ii) de `thermo` pour le calcul de l'énergie thermonucléaire libérée, iii) de `list` pour le calcul des flux de neutrinos. Au premier appel, il y a initialisation des abondances initiales et tabulation des réactions nucléaires suivant le cycle et la compilation des réactions à utiliser, *cf.* § 3.3 (Page 22). Lors des appels suivants, il y a calcul des variations des abondances ou/et de l'énergie thermonucléaire libérée, ou encore des flux de neutrinos.

Les isotopes utilisés dans les réseaux nucléaires dépendent, d'une part, du type de modèle à calculer et, d'autre part, des éléments chimiques auxquels on s'intéresse plus particulièrement.

La mise en œuvre de ces routines a bénéficié, à divers titres, de collaborations de B. Pichon, Y. Lebreton, S. Brun et D. Cordier. La source de CESAM2k contient diverses routines du type `nuc` spécialement adaptées à des cas spécifiques de l'évolution stellaire.

Description : On décrit l'organigramme d'une routine de type `nuc`, à l'exception de `iben`, *cf.* §8.60 (Page 171), et de `pp1`, *cf.* §8.96 (Page 191), dont les algorithmes diffèrent quelque peu du cadre général. Un argument d'entrée, `fait`, définit la partie du calcul à effectuer :

- `fait=1`
  - Initialisation du nombre d'isotopes du réseau nucléaire.
  - Initialisation du réseau nucléaire par appel à `rq_reac`, *cf.* §8.119 (Page 201).
  - Initialisation des abondances par appel à `abon_ini`, *cf.* §8.3 (Page 143).
  - Détermination du type et des caractéristiques de l'isotope fictif et calcul des abondances isotopiques initiales.
  - Initialisations diverses et écritures.
- `fait=2`
  - Calcul des taux de réaction.
  - Formation du jacobien.
- `fait=3`
  - Calcul de l'énergie nucléaire.
- `fait=4`
  - Calcul des flux de neutrinos. Les “flux sur terre” sont calculés en supposant que ces neutrinos sont produits à 1UA de la terre, même si le modèle n'est pas un modèle solaire.

Appel : `nuc` est appelée de divers endroits, en particulier de `rk_imps`, `thermo`, `eq_dif_chim`, `cesam`.

```
SUBROUTINE nuc(t,ro,comp,dcomp,jac,deriv,fait,
1 epsilon,et,ero,ex,hhe,be7e,b8e,n13e,o15e,f17e)
```

- Entrées :
  - `t, ro` : Température, densité.
  - `deriv=.TRUE.` : On calcule le jacobien et les dérivées.
  - `fait` : fait=1 : initialisation de la composition chimique, fait=2 : calcul des taux et du jacobien si deriv=.TRUE. fait =3 : énergie nucléaire et si `deriv=.TRUE.` : dérivées / température, densité et abondances, fait=4 : calcul des neutrinos.
- Entrées/Sorties :
  - `comp` : Abondances des isotopes par mole.
- Sorties :
  - `dcomp, jac` : Dérivées temporelles des abondances, jacobien.
  - `epsilon, et, ero, ex` : Energie thermonucleaire et dérivées.

- `hhe, be7e, b8e, n13e, o15e, f17e` : Nombres de neutrinos pour les réactions ^{1}H$(p, \beta^+\nu)^2$H, ^{7}Be$(e^-, \nu\gamma)^7$Li, ^{8}B$(\beta^+\nu)^8$Be ^{13}N$(\beta^+\nu)^{13}$C, ^{15}O$(\beta^+\nu)^{15}$N, ^{17}F$(\beta^+\nu)^{17}$O.

8.77 Routine générique `opa`

La routine PUBLIC `opa` du module `mod_opa`, *cf.* § E.6 (Page 259), est la ***routine générique*** qui, par l'intermédiaire de la routine spécifique dont le nom, `NOM_OPA`, est défini dans le fichier de données *cf.* § 3.3 (Page 22), réalise la gestion du calcul de l'opacité. Pour un triplet : (composition chimique, température, densité), les routines de type `opa` déterminent l'opacité Rosseland $\kappa(T, \rho, X, Z)$, ainsi que ses dérivées premières par rapport à i) la température, ii) la densité et, iii) la teneur en hydrogène. Deux types d'opacités sont implantés dans CESAM$2k$, des opacités analytiques et des opacités tabulées. Les premières sont des approximations, leur utilisation est robuste et ne demande que très peu de ressources ; peu précises elles sont réservées aux tests. Les secondes, plus précises, mais d'utilisation délicate, sont destinées aux exploitations.

Description : Les routines de type `opa`, la plupart d'origine externe, n'ont pas une structure commune ; chacune d'entre elles fait l'objet d'une description personnalisée. L'opacité conductive, *cf.* § 8.66 (Page 173), est introduite à l'issue du calcul de l'opacité radiative.

Appel : `opa` est appelée de divers endroits en particulier de `evol`, `thermo`, `thermo_atm`, `eq_dif_chim`, `cesam`.

```
SUBROUTINE opa(xh,t,ro,kappa,dkapdt,dkapdr,dkapdx)
```

- Entrées :
  - `xh, t, ro` : composition chimique, température, densité.
- Sorties :
  - `kappa,dkapdt,dkapdr,dkapdx` : opacité, dérivées.

8.78 Routine `opa_gong`

La routine PRIVATE `opa_gong` du module `mod_opa`, *cf.* § E.6 (Page 259), est une routine de type opacité analytique. Le formalisme utilisé est décrit dans le "Solar Model Comparison Project" de J.Christensen-Dalsgaard [1988]. L'opacité moyenne de Rosseland est approchée par une formule de Kramers améliorée, sous la forme d'une somme de deux termes, l'un valable pour l'intérieur stellaire et l'autre pour l'extérieur :

$$\kappa_{\rm R}^{-1} = \kappa_e^{-1} + \kappa_i^{-1}, \quad \kappa_e = c_e \rho^{m_e} T^{n_e}, \quad \kappa_i = c_i \rho^{m_i} T^{n_i},$$

avec : $c_e = 1.6236784 \times 10^{-33}$, $m_e = 0.407895$, $n_e = 9.28289$, $c_i = 7.1548412 \times 10^{13}$, $m_e = 0.138316$, $n_e = -1.97541$. Cette forme analytique correspond à $Z = 0.02$.

Appel : `opa_gong`, appelée par la routine générique `opa` *cf.* § 8.77 (Page 182), utilise les mêmes arguments.

8.79 Routine opa_houdek9

La routine PRIVATE `opa_houdek9` du module `mod_opa`, *cf.* § E.6 (Page 259), est une routine de type opacité tabulée. Elle utilise les interpolations des tables de Livermore 95 prolongées, pour les basses températures, par les opacités Kurucz ou Alexander ; elles peuvent, ou non, tenir compte de la conductivité électronique. Elles utilisent les bibliothèques et les données du package de Houdek & Rogl [1996]. La mise en forme des données est décrite au § 2.2.3 (Page 14). Pour son utilisation, il est nécessaire d'indiquer dans la routine `opa_houdek` le chemin qui permet d'atteindre les données `OPINPATH_95` ; il faut aussi indiquer au LINK le chemin de la bibliothèque `libopint.a`. La mise en œuvre de `opa_houdek` a bénéficié d'une collaboration avec N.Audard. La version originale du package de routines d'interpolation de G.Houdek se trouve à : `ftp ://solaris.tuwien.ac.at/incoming/`.

Description :
- Au premier appel :
  - Lecture et initialisation des tables.
- Adaptation et encadrement des données.
- Calcul de l'opacité, en cas d'échec appel à `opa_yveline_lisse`, *cf.* § 8.84 (Page 186).

Les routines d'interpolation d'origine externe ne sont pas décrites. Pour plus d'informations, se référer aux README du packetage "v9" du sous-directory SUN_STAR_DATA.

Appel : `opa_houdek9`, appelée par la routine générique `opa` *cf.* § 8.77 (Page 182), utilise les mêmes arguments.

8.80 Routine opa_int_zsx

La routine PRIVATE `opa__int_zsx` du module `mod_opa`, *cf.* § E.6 (Page 259), est une routine de type opacité tabulée. `opa_int_zsx` interpole les ***anciennes*** tables Livermore 1991, prolongées par les opacités Kurucz pour les basses températures. Elle utilise une interpolation spline 4D linéaire[3] suivant les axes $(T_6, \ln\rho/T_6^3, X, Z)$, $T_6 \equiv T \times 10^{-6}$. Les données Livermore 1991 sont actuellement caduques. Des difficultés de convergence, observées principalement dans la restitution de l'atmosphère, peuvent provenir de l'interpolation linéaire – non dérivable aux points de la table – dans ces tables aux larges intervalles tabulaires.

Description :
- Au premier appel :
  - Initialisation de constantes et extension des tables par interpolation.
  - Mise en forme des tables pour l'interpolation.
  - Estimations des limites des tables.
- Localisation du point de calcul.
- Interpolation et calcul des dérivées.

Appel : `opa_int_zsx`, appelée par la routine générique `opa` *cf.* § 8.77 (Page 182), utilise les mêmes arguments.

[3] La base utilisée est duale, les calculs d'interpolation sont alors immédiats.

8.81 Routine `opa_opal2`

La routine PRIVATE `opa_opal2` du module `mod_opa`, *cf.* § E.6 (Page 259), est une routine de type opacité tabulée. Elle permet de dépasser le seuil $Z < 0.1$ des données de OPAL. Elle tient compte des modifications de composition chimique dues à la nucléosynthèse $H \rightarrow He \rightarrow C \rightarrow N \rightarrow O$. Elle ne permet le calcul de l'opacité que pour des températures supérieures à 1eV, le raccord avec des opacités basse température n'existant pas. Elles n'est donc pas utilisable pour les modèles comportant une atmosphère. Le calcul de l'opacité est basé sur les tables d'opacité de type 2 de Livermore, il utilise les routines du paketage `z14xcotrin21`, *cf.* § 8.136 (Page 212), de A.I. Boothroyd. Les tables utilisées, placées dans le sous-directory `SUN_STAR_DATA` sont automatiquement décompressées et transformées en binaire à l'occasion de leur première utilisation.

`opa_opal2` ***est d'usage extrèmement délicat.***

Description :
- Au premier appel :
  - Identification du chemin directory où se trouvent les fichiers de données.
  - Identification des indices de C, N, O, estimation de l'abondance en masse des éléments de la mixture de GN93 qui ne sont pas pris en compte et du rapport [O/Fe].
  - Détermination des rapports C/Z, O/Z, pour les excès de C et O de la mixture utilisée par rapport à GN93.
  - Lecture en binaire des tables d'opacité concernées si elles existent, sinon création.
  - Calcul de X, Y, Z initiaux et définition des limites pour les sorties de table.
- Calcul de l'opacité.
- Estimation des dérivées par dérivation numérique.

Les routines d'interpolation, d'origine externe ne sont pas décrites, *cf.* § 8.136 (Page 212).

Appel : La liste d'appel de cette routine est celle de sa routine générique `opa` *cf.* § 8.77 (Page 182), avec un argument supplémentaire, la variable logique `cno` qui permet une alternative :
- `cno=.FALSE.` : seules les variations de l'abondance en carbone sont prises en compte.
- `cno=.TRUE.` : les variations de l'abondance en carbone et en oxygène sont prises en compte.

En raison de l'absence d'un raccord avec des opacités basse température, `opa_opal2` est utilisée pour prolonger les routines d'opacité `opa_yveline`, *cf.* § 8.83 (Page 185), et `opa_yveline_lisse`, *cf.* § 8.84 (Page 186), au delà de $Z > 0.1$.

Il est toutefois possible d'utiliser `opa_opal2` directement, en codant dans le fichier de données, *cf.* § 3.3 (Page 22), `NOM_OPA=opa_opal2_cno`, pour tenir compte seulement des variations de l'abondance du carbone, ou `NOM_OPA=opa_opal2_co`, pour tenir compte des variations de l'abondance du carbone et de l'oxygène.

8.82 Routine opa_opalCO

La routine PRIVATE opa_opalCO du module mod_opa, *cf.* § E.6 (Page 259), est du type opacité tabulée. Elle permet de dépasser le seuil $Z < 0.1$ des données de OPAL. Elle tient compte des modifications de composition chimique dues à la nucléosynthèse $H \rightarrow He \rightarrow C \rightarrow O$. Elle ne permet le calcul de l'opacité que pour des températures supérieures à 1eV, le raccord avec des opacités basse température n'existant pas. Elle est inutilisable pour les modèles comportant une atmosphère.

Les tables utilisées par ces routines, placées dans le sous-directory SUN_STAR_DATA sont automatiquement décompressées et transformées en binaire à l'occasion de leur première utilisation.

La mise en œuvre de opa_opalCO a bénéficié d'une collaboration avec L.Piau.

Description :
- Au premier appel :
  - Identification des indices de H, He, C, O.
  - Définition de la métallicité de la table et calcul des rapports C/Z, O/Z.
- Détermination des abondances de H, He, C, O, et des excès de C et O.
- Calcul des opacités.

opa_opalCO utilise une interpolation moins sophistiquée des tables OPAL que opa_opal2. Les routines d'interpolation utilisées sont des routines subordonnées de opa_opalCO ; d'origine externe, ces routines ne sont pas décrites. Il s'agit de :
opac, cointerp, t6rinterp, readco, quad, spline, splint, fity, fitx, getd, interp, smooth, opaltab.

Appel : opa_opalCO, appelée par la routine générique opa *cf.* § 8.77 (Page 182), utilise les mêmes arguments.

8.83 Routine opa_yveline

La routine PRIVATE opa_yveline du module mod_opa, *cf.* § E.6 (Page 259), est une routine de type opacité tabulée. L'interpolation du package construit par Y. Lebreton pour le calcul de l'opacité OPAL, Iglesias & Rogers [1991]. opa_yveline effectue une interpolation cubique dans la table, linéaire aux bords.La table en ASCII se trouve dans le fichier opa_yveline_etendu.dat du sous-directory SUN_STAR_DATA. Extraire cette table et la transformer en un fichier binaire à l'aide du programme ascii2bin_opa de ce package *cf.* § 2.2.1 (Page 13).
Lorsque l'abondance en éléments lourds dépasse $Z > 0.1$ opa_yveline fait appel à opa_opal2, *cf.* § 8.81 (Page 184), qui permet de tenir compte de la nucléosynthèse $H \rightarrow He \rightarrow C \rightarrow N \rightarrow O$.

Description :
- Au premier appel :
  - Vérification de la cohérence des données.
- Vérification de l'accessibilité du point d'interpolation.
- Calcul de l'opacité si $Z < 0.1$.
- Appel à opa_opal2 si $Z > 0.1$.

Les routines d'interpolation utilisées par `opa_yveline` sont introduites sous forme de routines subordonnées, d'origine externe, elles ne sont pas décrites. Il s'agit de : `kappa_opal, intlin_opal, intl_opal, sub_quad, lect_opal, lpol_op, pos_table_op.`

Appel : `opa_yveline`, appelée par la routine générique `opa` *cf.* § 8.77 (Page 182), utilise les mêmes arguments.

8.84 Routine `opa_yveline_lisse`

La routine PRIVATE `opa_yveline_lisse` du module `mod_opa`, *cf.* § E.6 (Page 259), est une routine de type opacité tabulée. L'exploitation du package construit par Y. Lebreton pour le calcul de l'opacité OPAL, Iglesias & Rogers [1991], et effectué par `opa_yveline_lisse` est un lissage par des Béziers, dont l'ordre peut être facilement adapté pour chacune des dimensions (X, Z, T, ρ). Par défaut l'ordre 4 est utilisé, l'ordre 2 correspond à l'interpolation linéaire. Le lissage assure des dérivées d'ordre élevé, il est plus robuste que l'interpolation. Le prix à payer est une précision réduite. La table en ASCII se trouve dans le fichier `opa_yveline_etendu.dat` du sous-directory `SUN_STAR_DATA`. Extraire cette table et la transformer en un fichier binaire à l'aide du programme `ascii2bin_opa` de ce package *cf.* § 2.2.1 (Page 13).

Description :
- Au premier appel :
  - Lecture des tables.
  - Définitions diverses.
- Vérification de l'accessibilité du point d'interpolation.
- Appel à `opa_opal2` si $Z > 0.1$.
- Si $Z < 0.1$, recherche des indices, calcul de l'opacité.

Appel : `opa_opa_yveline_lisse`, appelée par la routine générique `opa` *cf.* § 8.77 (Page 182), utilise les mêmes arguments.

8.85 Routine `osc_adia, osc_invers, osc_noad`

Ces routines PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), de type `output`, *cf.* § 8.86 (Page 187), ont pour fonction la création des fichiers utilisés respectivement pour le calcul des oscillations adiabatiques, des inversions et des oscillations non adiabatiques.

Description :
Les tableaux d'indices `tglob(nglob)` et `tvar(nvar)`, adaptés à chaque fichier de sortie, définissent l'ordre dans lequel sont disposés les éléments de ces fichiers ASCII. Les éléments des tableaux `var(P, T, L...)` et `glob(age, d2ro...)` construits dans la routine `cesam`, *cf.* § 8.10 (Page 147), et complétés dans la routine `add_ascii`, *cf.* § 8.4 (Page 143), sont réorganisés dans les tableaux intermédiaires `eglob(nglob)` et `evar(nvar,itot)`, suivant l'ordre défini dans `tglob(nglob)` et `tvar(nvar)`. Le fichier de sortie est ensuite créé à partir des fichiers intermédiaires. L'identification

des quantités est décrite au § D.1 (Page 239). Un exemple de création de tels fichiers est décrit au § 4.7 (Page 42).

Appel : `osc_adia`, `osc_invers`, `osc_noad`, appelées par la routine générique `output` *cf.* § 8.86 (Page 187), utilisent les mêmes arguments.

8.86 Routine générique `output`

Cette routine PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), est la ***routine générique*** des routines de formation des fichiers ASCII de sortie, utilisés pour les calcul des oscillations ou des inversions, *cf.* § 8.85 (Page 186), ou encore pour créer un fichier de sortie ASCII personnalisé, *cf.* § 4.7 (Page 42).

Description :
Suivant le type de sortie ASCII indiqué par la variable `NOM_OUTPUT` du fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), la routine de formation du fichier ASCII est appelée.

Appel : `output` est appelée par la routine `ecrit_ascii`, *cf.* § 8.42 (Page 161) subordonnée de la routine `cesam` ; `output` n'a que des arguments d'entrée.

```
SUBROUTINE output(var,glob,itot,ivar)
```

- Entrées :
  - `glob, itot, ivar` : quantités globales, nombre total de points, nombre de variables

8.87 Routine générique `pertm`

Cette routine PUBLIC du module `mod_static`, *cf.* § E.12 (Page 263), est la ***routine générique*** du calcul de la perte de masse.

Description :
Suivant le type de sortie ASCII indiqué par la variable `NOM_PERTM` du fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), la routine de perte de masse est appelée.

Appel : `pertem` est appelée par `evol`, *cf.* § 8.55 (Page 167), et `resout`, *cf.* § 8.114 (Page 195).

```
SUBROUTINE pertem(dt))
```

- Entrée
  - `dt` : pas temporel

8.88 Routine `pertm_ext, pertm_msol`

Pour ces deux routines PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), la perte de masse est supposée linéaire en fonction du temps. Le taux `mdot`$M_\odot$ par an est celui indiqué dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22).

Avec `pertm_msol`, le taux de perte de masse ainsi que le traitement du vent *cf.* § 4.5 (Page 40), sont annulés dès que la masse du modèle atteint la masse solaire.

Description :
- Extraction des masses.
- Formation du tableau des masses corrigées de la perte/gain de masse.

Appel : ces deux routines ont les mêmes arguments que leur routine générique `pertm`, *cf.* § 8.87 (Page 187).

8.89 Routine pertm_tot

Cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), tient compte de la masse perdue par la transformation de la masse en énergie, par les réactions thermonucléaires. La perte de masse externe est supposée linéaire en fonction du temps. Le taux `mdot`$M_\odot$ par an est celui indiqué dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22).

Description :
- Extraction des masses.
- Calcul de la perte de masse locale due aux réactions nucléaires.
- Formation du tableau des masses corrigées de la perte/gain de masse totale.

Appel : cette routine a les mêmes arguments que sa routine générique `pertm`, *cf.* § 8.87 (Page 187).

8.90 Routine `pertem_waldron`

Cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), utilise la perte de masse empirique de Waldron, citée dans l'article A.Ap 229 (1990) 469-474.

Description :
- Extraction des masses.
- Calcul de la perte de masse.
- Formation du tableau des masses corrigées de la perte de masse.

Appel : cette routine a les mêmes arguments que sa routine générique `pertm`, *cf.* § 8.87 (Page 187).

8.91 Routine générique `pertw`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), est la ***routine générique*** du calcul de la perte/gain de moment cinétique.

Description :
Suivant la description de la perte de moment cinétique indiquée par la variable `NOM_PERTW` du fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), la routine spécifique de perte de moment cinétique est utilisée.

Appel : `pertw` est appelée par `coeff_rota`, *cf.* § 8.13 (Page 149).

```
SUBROUTINE pertw(nu,omega,r,mw_dot)
```

- Entrées :
  - `nu, omega, r` : masse, vitesse angulaire, rayon au point de calcul.
- Sortie
  - `mw_dot` : taux de perte/gain de moment cinétique.

8.92 Routine `pertw_loc`

Avec cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), de type `pertw`, la perte / gain de moment cinétique est proportionnelle à l'énergie cinétique de rotation locale et supposée localisée dans la fraction de masse externe `M > pw_extend`, *cf.* § 7.9.18 (Page 125).

Description :
La variation temporelle de moment cinétique par unité de masse est modélisée par :

$$\dot{\mathcal{M}}_\Omega = \gamma R^2\Omega^2, \quad |\gamma| \sim 10^{-13} \sim 10^{-14}$$

Le paramètre γ correspond au taux `p_pertw` indiqué dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), le taux de perte/gain de moment cinétique `mw_dot` est nul en deça de la limite fixée.

Appel : cette routine utilise les mêmes arguments que sa routine générique `pertw`, *cf.* § 8.91 (Page 188).

8.93 Routine `pertw_ptm`

Avec cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), de type `pertw`, le taux `p_pertw` de perte de moment cinétique indiqué dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), est utilisé comme un paramètre fixant l'efficacité de la perte de moment cinétique résultant de la perte de masse, *cf.* § 7.9.18 (Page 125). La perte de moment cinétique est supposée localisée dans la fraction de masse externe `M > pw_extend`.

Description :
L'apport/perte de moment cinétique par unité de masse est approché par :

$$\dot{\mathcal{M}}_\Omega = \frac{\dot{\mathcal{M}}}{m_{\rm rot}} R^2\Omega_{\rm p}.$$

Le taux de perte/gain de moment cinétique `mw_dot` est fixé à 0 en deça de la limite fixée.

Appel : cette routine utilise les mêmes arguments que sa routine générique `pertw`, *cf.* § 8.91 (Page 188).

8.94 Routine pertw_sch

Avec cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), de type `pertw`, le taux `p_pertw` de perte de moment cinétique indiqué dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), est utilisé comme un paramètre fixant l'efficacité d'une perte de moment cinétique proportionnelle à Ω^3 (Schumanish), *cf.* § 7.9.18 (Page 125).

Description :
La variation temporelle de moment cinétique par unité de masse est modélisée par :

$$\dot{\mathcal{M}}_\Omega = aR^2\Omega_{\rm s}^3, \quad |a| \sim 1.10^{-9}{\rm s}.$$

Le taux de perte de moment cinétique `mw_dot` est fixé à 0 en deça de la limite fixée.

Appel : cette routine utilise les mêmes arguments que sa routine générique `pertw`, *cf.* § 8.91 (Page 188).

8.95 Routine planetoides

CESAM2k offre la possibilité de tenir compte des variations de composition chimique dues à des chutes de planétoïdes dans la zone convective externe, *cf.* § 7.8.2 (Page 99). Cette possibilité est gérée par la routine PUBLIC `planétoides` du module `mod_nuc`, *cf.* § E.9 (Page 261).

Description :
- Lors d'un appel d'initialisation, depuis la routine `evol`, on effectue une recherche dans l'environnement du fichier où sont indiquées les fractions de masse des planétoïdes, *cf.* § 4.6 (Page 41). Si un tel fichier est trouvé, les fractions de masse sont normalisées, puis réparties entre les différents isotopes du réseau nucléaire, et enfin transformées en fraction par mole ; la variable logique `l_planet` est déclarée `.TRUE.`
- Lors des appels suivants, une correction aux taux de réactions nucléaires est effectuée lorsque l'âge du modèle est dans l'intervalle de temps des chutes. *cf.* § 7.8.2 (Page 99).

Appel : `planetoides` est appelée par :
- `evol`, *cf.* § 8.55 (Page 167), pour initialisation.
- `rk_imps`, *cf.* § 8.117 (Page 199) ou `eq_dif_chim`, *cf.* § 8.38 (Page 159), à la suite d'un appel à la routine `nuc`, *cf.* § 8.76 (Page 179).

Les arguments sont optionnels.

```
SUBROUTINE planetoides(xchim,dxchim,jac,m_planet,mw_planet)
```

- Entrées :
  - `xchim` : Composition chimique par mole.
- Entrées/Sorties :
  - `dxchim, jac` : Taux de réactions nucléaires et jacobien.
  - `m_planet,mw_planet` : taux d'accroissement de masse et de moment cinétique.

8.96 Routine `pp1`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***test***, *cf.* § 8.76 (Page 179). Selon les "*Computational procedures for GONG solar model project*" (Christensen-Dalsgaard [1988]), seul l'isotope ^{1}H est explicitement pris en compte. L'abondance initiale d'hydrogène, le taux de la réaction et l'énergétique sont calculés *in situ* par une approximation analytique du cycle PP.

Toutefois la structure générale des routines de type `nuc` est respectée.

Appel : `pp1` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.97 Routine `pp3`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***test***, *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{3}He, ^{4}He, et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Li, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine, qui ne concerne que le cycle PP, a pour vocation de servir pour les tests de mise au point.

Appel : `pp3` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.98 Routine `ppcno10`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3***α, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{7}Li, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la diffusion microscopique du lithium, en tenant compte de l'évolution au long de la pré-séquence principale.

Appel : `ppcno10` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.99 Routine `ppcno10BeBFe`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3***α, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{6}Li, ^{7}Li, ^{9}Be, ^{11}B, ^{12}C, ^{13}C, ^{14}N,^{15}N, ^{16}O,^{17}O, ^{56}Fe et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la diffusion microscopique des éléments légers et du fer, en tenant compte de l'évolution au long de la pré-séquence principale.

Appel : `ppcno10BeBFe` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.100 Routine `ppcno10Fe`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3***α, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{7}Li, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O, ^{56}Fe et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la diffusion microscopique du lithium et du fer.

Appel : `ppcno10Fe` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.101 Routine `ppcno10K`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3***α, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{7}Li, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O et deux éléments fictifs ExK et Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136). Les deux éléments fictifs représentent respectivement les éléments de masse inférieure et supérieure au potassium. Cette routine a été construite pour décrire la diffusion microscopique différentielle entre les éléments de masse supérieure ou inférieure au potassium.

Appel : `ppcno10K` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.102 Routine `ppcno11`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3***α, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{2}H, ^{3}He, ^{4}He, ^{7}Li, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O et l'élément fictif Ex, sont explicitement pris en compte ; ^{7}Be est supposé à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la pré-séquence principale, en évitant de suivre l'abondance du ^{7}Be dont l'échelle de temps d'évolution est infime.

Appel : `ppcno11` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.103 Routine `ppcno12`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3***α, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{2}H, ^{3}He, ^{4}He, ^{7}Li, ^{7}Be, ^{12}C, ^{13}C, ^{14}N,^{15}N, ^{16}O,^{17}O et l'élément fictif Ex, sont explicitement pris en compte ; Aucun isotope n'est à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire l'évolution sans élément à l'équilibre.

Appel : `ppcno12` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.104 Routine `ppcno12Be`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3α***, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{2}H, ^{3}He, ^{4}He, ^{7}Li, ^{7}Be, ^{9}Be, ^{12}C, ^{13}C, ^{14}N,^{15}N, ^{16}O,^{17}O et l'élément fictif Ex, sont explicitement pris en compte ; Aucun isotope n'est à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la diffusion microscopique du beryllium en tenant compte de l'évolution au long de la pré-séquence principale.

Appel : `ppcno12Be` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.105 Routine `ppcno12BeBFe`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3α***, *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{2}H, ^{3}He, ^{4}He, ^{6}Li, ^{7}Li, ^{7}Be, ^{9}Be, ^{11}B, ^{12}C, ^{13}C, ^{14}N,^{15}N, ^{16}O,^{17}O, ^{56}Fe et l'élément fictif Ex, sont explicitement pris en compte. Aucun isotope n'est à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la diffusion microscopique des éléments légers, et du fer, en tenant compte de l'évolution au long de la pré-séquence principale.

Appel : `ppcno12BeBFe` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.106 Routine `ppcno12Li`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3α***, *cf.* § 8.76 (Page 179).

Les isotopes ^{1}H, ^{2}H, ^{3}He, ^{4}He, ^{6}Li, ^{7}Li, ^{7}Be, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O,^{17}O et l'élément fictif Ex, sont explicitement pris en compte. Cette routine a été construite pour décrire la diffusion microscopique des éléments légers, plus particulièrement du lithium, en tenant compte de l'évolution au long de la pré-séquence principale. Aucun isotope n'est à l'équilibre, *cf.* § 7.11.4 (Page 136).

Appel : `ppcno12Li` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.107 Routine `ppcno3a12Ne`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***3α***, *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N,^{16}O,^{17}O, ^{18}O, ^{20}Ne, ^{22}Ne et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Li, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136).

Appel : `ppcno3a12Ne` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.108 Routine `ppcno3a9`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***3*** α, *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N,^{16}O,^{17}O et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Li, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136).

Appel : `ppcno3a9` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.109 Routine `ppcno3ac10`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***CO***, *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N,^{16}O,^{17}O, ^{20}Ne et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Li, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136).

Appel : `ppcno3ac10` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.110 Routine `ppcno9`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3*** α *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N,^{16}O,^{17}O et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Li, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136).

Appel : `ppcno9` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.111 Routine `ppcno9Fe`

Cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est de type `nuc`, catégorie ***pré-3*** α, *cf.* § 8.76 (Page 179). Les isotopes ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{15}N,^{16}O,^{17}O, Fe56 et l'élément fictif Ex, sont explicitement pris en compte ; ^{2}H, ^{7}Li, ^{7}Be sont supposés à l'équilibre, *cf.* § 7.11.4 (Page 136). Cette routine a été construite pour décrire la diffusion microscopique du fer.

Appel : `ppcno9Fe` appelée par la routine générique de réactions nucléaires `nuc` a la même liste d'appel.

8.112 Routine print_ctes

Cette routine PUBLIC du module `mod_donnees`, *cf.* § E.3 (Page 248), a pour fonction la transcription sur l'unité d'écriture des valeurs des principales constantes physiques utilisées.

Appel : `print_ctes` est appelée de `cesam`, *cf.* § 8.10 (Page 147)

```
SUBROUTINE print_ctes(i)
```

- Entrée
  - `i` : indice de l'unité FORTRAN sur laquelle l'écriture est effectuée.

8.113 Routine read_ascii

Cette routine PUBLIC du module `mod_exploit`, *cf.* § E.14 (Page 264), a pour fonction la lecture des fichiers ASCII d'oscillation, *cf.* § 8.85 (Page 186).

Description
- Ouverture du fichier de type `mon_modele.osc`, lecture de l'en-tête et des constantes.
- Lecture des variables.

Appel : `read_osc` est appelée par divers programmes de dessin, en particulier `des_osc`, `des_abon`, `des_diff_osc` ou encore `lit_osc`.

```
SUBROUTINE read_ascii(nom_fich,itot,nglob,nvar,abid)
```

- Entrée
  - `nom_fich` : nom du fichier de type `mon_modele.osc` (sans l'extension `.osc`).
- Sorties :
  - `itot, nglob, nvar, abid` : nombre de points, de "constantes global", de variables du tableau `var`, sans les éléments chimiques,

La description des quantités extraites de ces fichiers est donnée au § D.1 (Page 239).

8.114 Routine resout

La fonction de la routine PRIVATE `resout` du module `mod_static`, *cf.* § E.12 (Page 263), est la gestion de la résolution des équations de l'évolution stellaire pour :

1. Le modèle en cours d'évolution.
2. Le modèle de séquence principale d'âge zéro homogène.
3. Le modèle initial de pré-séquence principale homogène.

La figure 8.4 (Page 196), présente un organigramme de la routine `resout`.

Description : L'architecture des algorithmes de résolution des équations de la structure interne est assez similaire dans les trois phases de l'évolution évoquées çi-avant. La structure de chacune d'entre elles est celle d'une itération Newton-Raphson.
- Initialisation de divers paramètres généraux.
- Poursuite d'une évolution :

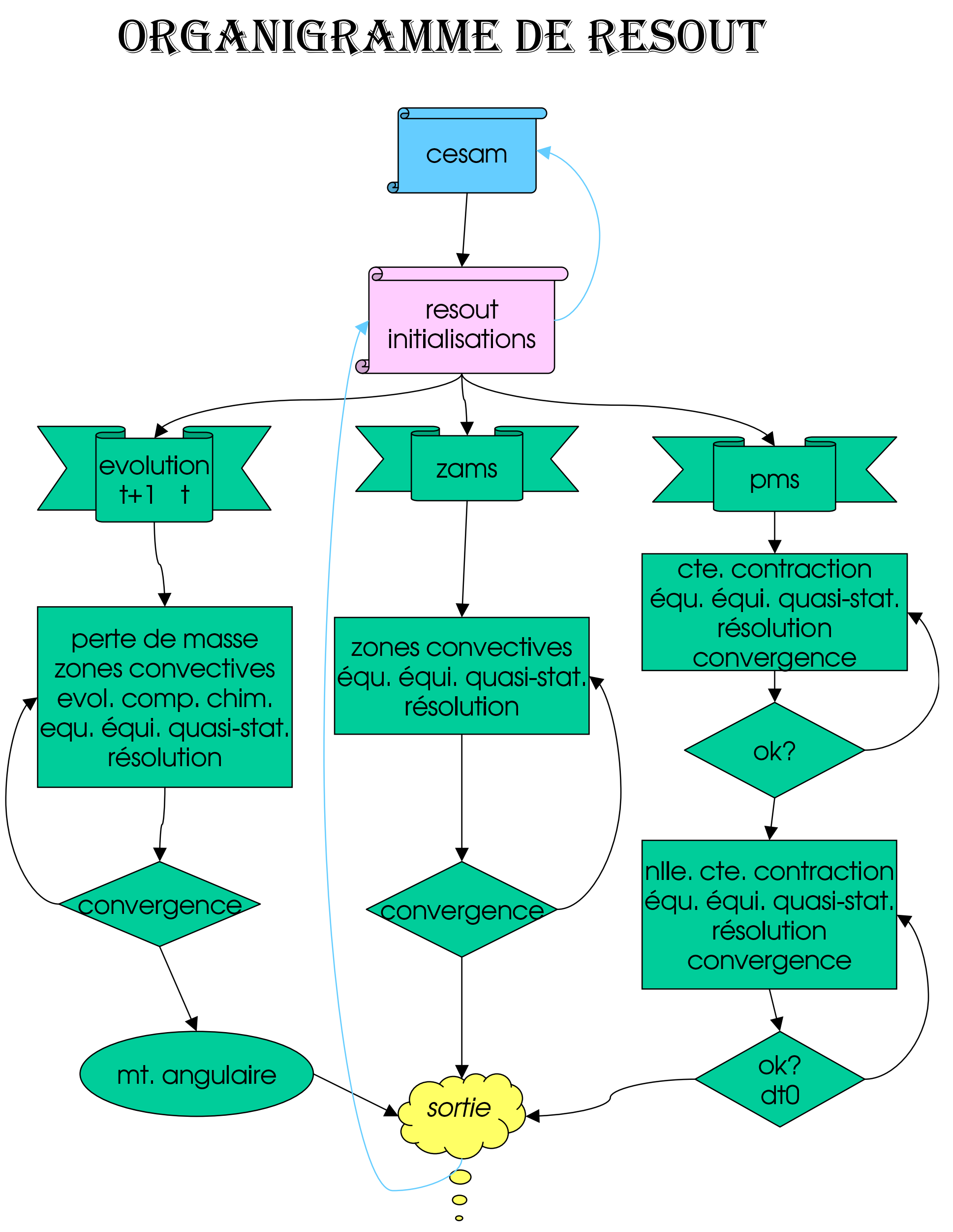

FIG. 8.4 – Organigramme de la routine de la gestion de la résolution des équations de la structure interne `resout`.

  - Ajustement de paramètres s'il s'agit du calcul du modèle ultime.
  - Appel à `update`, *cf.* § 8.131 (Page 210) pour mise en mémoire du modèle au pas temporel précédent.
  - Initialisation de compteurs et début de l'itération Newton-Raphson.
  - Calcul de la perte de masse.
  - Estimation du nombre de couches nécessaire pour assurer à la constante de répartition la valeur définie par la précision requise, *cf.* § 7.2.4 (Page 70).
  - Appel à `lim_zc`, *cf.* § 8.69 (Page 174), pour un réaménagement éventuel de la discrétisation spatiale, la détection des limites zones radiatives / zones convectives et l'ajustement des facteurs de répartition, *cf.* § 7.2.5 (Page 73).
  - Evolution de la composition chimique, éventuellement du moment cinétique, par appel à `evol`, *cf.* § 8.55 (Page 167).
  - Formation et résolution des équations de l'équilibre quasi-statique par appel à `coll_qs`, *cf.* § 8.18 (Page 151).
  - Suivant l'estimation de la précision :
    - Ajustement du pas temporel pour satisfaire les conditions d'arrêt particulières, réinitialisation et reprise des itérations Newton-Raphson.
    - Poursuite des itérations Newton-Raphson.
    - Réinitialisation avec diminution du pas temporel et reprise.
    - Gestion éventuelle de la rotation solide, estimation de l'énergie graviphique, tabulation de diverses quantités, retour vers `cesam`, *cf.* § 8.10 (Page 147).
    - Appel à `sortie`, *cf.* § 8.121 (Page 203) pour arrêt du calcul.
- Modèle initial de séquence principale d'âge zéro homogène, ou de pré-séquence principale homogène :
  - Estimation du nombre de couches nécessaires pour assurer à la constante de répartition la valeur définie par la précision requise, *cf.* § 7.2.4 (Page 70).
  - Appel à `lim_zc`, *cf.* § 8.69 (Page 174), pour un réaménagement éventuel de la discrétisation spatiale, la détection des limites zones radiatives / zones convectives et l'ajustement des facteurs de répartition, *cf.* § 7.2.5 (Page 73).
  - Initialisation de compteurs et début de l'itération Newton-Raphson.
  - Formation et résolution des équations de l'équilibre quasi-statique par appel à `coll_qs`, *cf.* § 8.18 (Page 151).
  - Suivant l'estimation de la précision :
    - Poursuite des itérations Newton-Raphson, précédée par une réactualisation de la localisation des limites zones radiatives / zones convectives par appel à `lim_zc`.
    - Gestion éventuelle de la rotation solide, estimation de l'énergie graviphique, tabulation de diverses quantités, retour vers `cesam`, *cf.* § 8.10 (Page 147).
    - Appel à `sortie`, *cf.* § 8.121 (Page 203), pour arrêt du calcul.

Appel : à diverses reprises `resout` est appelée par `cesam`, *cf.* § 8.10 (Page 147).

```
SUBROUTINE resout(un23,dt,dts)
```

- Entrées :
  - `un23,dt,` : Paramètre du type de modèle, pas temporel.

- Sorties :
  - `dts` : Estimation de la valeur du pas temporel suivant.

8.115 Routine `resout_chim`

Cette routine PRIVATE du module `mod_evol`, *cf.* §E.11 (Page 262), a pour fonction la résolution par éléments finis du système implicite des équations de la diffusion des espèces chimiques, *cf.* §7.8 (Page 96). Les produits scalaires sont approchés par une intégration numérique de type Gauss, suivant un algorithme inspiré de l'algorithme 5.22 p.203 de Schumaker [1981]. De structure identique à `resout_rota`, *cf.* §8.116 (Page 198), ces deux routines pourraient être rassemblées en une seule, elles sont distinctes pour faciliter la lisibilité.

Description : L'architecture globale de la routine est celle d'une itération Newton-Raphson. Les coefficients provisoires des B-splines, *i.e.* les inconnues, sont initialisés dans la routine d'appel `diffus`, *cf.* §8.38 (Page 159).

- Initialisations au premier appel.
- Définition du rang du système, du nombre de lignes, des indices de première colonne, initialisation de compteurs.
- Itérations Newton-Raphson :
  - Pour chaque intervalle du vecteur nodal, détermination des abscisses de Gauss, formation des coefficients des équations par appel à `eq_diff_chim`, *cf.* §8.46 (Page 163).
  - Pour chaque B-spline non identiquement nulle sur l'intervalle, évaluation du second membre du système linéaire et de sa contribution au jacobien.
  - Résolution du système linéaire bande, correction des coefficients provisoires des B-splines, Estimation de la précision.
- Suivant la précision obtenue, poursuite des itérations Newton-Raphson, retour vers la routine appelante, réinitialisation si la précision requise n'a pas été atteinte après un nombre fixé d'itérations.

Appel : `resout_chim` est appelée par `diffus`, *cf.* §8.38 (Page 159).

```
SUBROUTINE resout_chim(dt,ok)
```

- Entrées :
  - `dt` : Pas temporel.
- Sorties :
  - `ok=.TRUE.` : La convergence a été obtenue.

8.116 Routine `resout_rota`

Cette routine PRIVATE du module `mod_evol`, *cf.* §E.11 (Page 262), a pour fonction la résolution par éléments finis du système implicite des équations de la diffusion du moment cinétique, *cf.* §7.9 (Page 111). Les produits scalaires sont approchés par une intégration numérique de type Gauss, suivant un algorithme inspiré de l'algorithme 5.22 p.203 de Schumaker [1981]. De structure identique à `resout_chim`, *cf.* § 8.115 (Page 198), ces deux routines pourraient être rassemblées en une seule, elles sont distinctes pour faciliter la lisibilité.

Description : L'architecture globale de la routine est celle d'une itération Newton-Raphson. Les coefficients provisoires des B-splines, *i.e.* les inconnues, sont initialisés dans la routine d'appel `diffus`, *cf.* § 8.38 (Page 159).

- Initialisations au premier appel.
- Définition du rang du système, du nombre de lignes, des indices de première colonne, initialisation de compteurs.
- Itérations Newton-Raphson :
  - Pour chaque intervalle du vecteur nodal, détermination des abscisses de Gauss, formation des coefficients des équations par appel à `eq_diff_rota`, *cf.* § 8.47 (Page 164).
  - Pour chaque B-spline non identiquement nulle sur l'intervalle, évaluation du second membre du système linéaire et de sa contribution au jacobien.
  - Pour le centre, formation des coefficients des parties intégrées par appel à `eq_diff_rota`, *cf.* § 8.47 (Page 164), et calcul des contributions au second membre et au jacobien du système linéaire.
  - Pour la surface, formation des coefficients des parties intégrées par appel à `eq_diff_rota`, *cf.* § 8.47 (Page 164), et calcul des contributions au second membre et au jacobien du système linéaire.
  - Résolution du système linéaire bande, correction des coefficients provisoires des B-splines, estimation de la précision.
- Suivant la précision obtenue, poursuite des itérations Newton-Raphson, retour vers la routine appelante, réinitialisation si la précision requise n'a pas été atteinte après un nombre fixé d'itérations.

Appel : `resout_rota` est appelée par `diffus`, *cf.* § 8.38 (Page 159).

```
SUBROUTINE resout_rota(dt,ok)
```

- Entrées :
  - `dt` : Pas temporel.
- Sorties :
  - `ok=.TRUE.` : La convergence a été obtenue.

8.117 Routine `rk_imps`

Cette routine PRIVATE du module `mod_evol`, *cf.* § E.11 (Page 262), a pour fonction l'intégration des équations de l'évolution des espèces chimiques *e.g.* Eq. 7.98 (Page 138) en l'absence de diffusion. Les échelles de temps d'évolution des divers isotopes diffèrent par plusieurs dizaines de magnitudes. A titre d'exemple, dans le soleil, l'échelle de temps d'évolution de ^{2}H est de quelques minutes, alors que celle de ^{1}H est de l'ordre de la centaine de million d'années. La disparité de ces échelles de temps est à l'origine de difficultés numériques : si on s'impose de suivre la plus courte échelle de temps, le temps de calcul est inimaginable, si on s'impose de suivre la plus grande échelle de temps, les erreurs numériques engendrées sur les espèces chimiques avec courte échelle de temps font "exploser" le calcul.

Pour ce type de problème numérique, qualifié de "raide", ou encore de "mal posé", il existe des algorithmes spécifiques qui, *en gros*, permettent de suivre l'échelle de temps que l'on désire, *e.g.* celle de ^{1}H pour le soleil, sans que, pour autant, les erreurs numériques engendrées sur les espèces chimiques à courte échelle de temps ne perturbent la solution de façon catastrophique. Ces algorithmes sont implicites, ils nécessitent le calcul d'un Jacobien ; en cas de non linéarité, la solution est obtenue par approximations successives, le plus souvent par la méthode de Newton-Raphson, qu'il convient parfois d'aménager.

Dans le cas de la structure interne, le mélange convectif complique encore le problème qui, non seulement est raide, mais est donc aussi intégral. En résumé, le problème du calcul de l'évolution des espèces chimiques est un problème intégro-différentiel aux valeurs initiales, non linéaire, raide.

La routine `rk_imps` utilise le formalisme de Runge-Kutta implicite avec le schéma Lobatto IIIC, *cf.* Hairer & Wanner [1991], ainsi qu'il a été décrit au § 7.5.3 (Page 87), noté dans la suite IRK Lobatto IIIC. L'ordre d'intégration est limité au troisième ordre. L'utilisation d'un ordre supérieur nécessiterait l'interpolation de la température, de la densité et de la composition chimique en tous les points du modèle entre différents pas temporel, interpolations dans lesquelles serait perdu le gain en précision que l'on peut attendre de l'utilisation d'un schéma d'ordre élevé. Cette routine originale est l'une des plus "techniques" et élégantes de CESAM2*k*.

Choix du pas temporel :

La valeur du pas temporel à utiliser dépend de la précision requise. A l'issue de chaque intégration, la précision obtenue est estimée. Dans le programme appelant `evol`, *cf.* § 8.55 (Page 167), le pas temporel est alors diminué (*respt.* augmenté) si la précision obtenue est inférieure (*respt.* supérieure) à ce qui est souhaité. La précision d'une intégration peut être estimée en l'effectuant avec des algorithmes différents, ou des pas temporels différents, ce qui nécessite de refaire plusieurs fois le calcul. Pour limiter l'effort de calcul, avec `rk_imps` on se contente de contrôler le pas temporel en limitant les variations de certains isotopes au cours d'un pas temporel. Cette méthode, quoique très répandue, n'est pas correcte car une grande (*respt.* petite) variation d'une abondance ne signifie pas, ***nécessairement***, que l'intégration est imprécise (*respt.* précise).

Description :

- Initialisations des matrices de IRK Lobatto IIIC suivant l'ordre de précision, `ordre`, défini dans la routine `cesam`, *cf.* § 8.10 (Page 147), des contraintes du schéma de Newton-Raphson et de l'ordre de grandeur de chaque abondance utilisée pour estimer la précision relative de l'intégration et de la convergence.
- Pour chaque temps intermédiaire, appel à la routine de réactions thermonucléaires pour le calcul des Ψ_i, éventuellement appel à la routine `vent`, *cf.* § 4.5 (Page 40), pour tenir compte de la perte ou du gain de masse, calcul de la contribution à la ligne correspondante du jacobien, détermination de la contribution à l'intégrale de Gauss dans les zones convectives.
- Formation de la ligne correspondante de Eq. 7.23 (Page 89).
- Solution du sytème linéaire donnant les corrections du processus itératif.
- Estimation de la précision et des nouvelles abondances, et poursuite, si besoin, du processus itératif.

Arguments de `rk_imps`

```
SUBROUTINE rk_imps(t_t,ro_t,compx,t,ro,compy,dt,esti,ok,nuc,kk,z_vent,dm)
```

– Entrées :
  – `t_t, ro_t, t, ro` : tables des températures, densité aux temps t et $t+dt$;
  – `dt, kk, dm, compx, z_vent` : pas temporel, nombre de couches à mélanger, tables des intervalles de masse et de la composition chimique au temps t, variable logique indiquant que l'on doit corriger les taux de réactions nucléaires, pour tenir compte d'un vent de composition chimique différente de celle des couches extrernes *cf.* § 8.133 (Page 210).
– Entrées/Sorties :
  – `compy` : table des la composition chimique au temps $t+dt$.
– Sorties :
  – `ok=.TRUE.` : la précision requise est atteinte.

Appel : `rk_imps` est appelée par `evol`.

8.118 Routines `roger00, roger02, roger05, roger10a`

Ces routines PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), constituent des lois $T(\tau]$ dérivées par R.Cayrel du programme ATLAS12 de Kurucz. Elles diffèrent entre elles par la métallicité de la mixture de type ***solaire*** utilisée :
– `roger00` : [Fe/H]=0.0.
– `roger02` : [Fe/H]=-0.2.
– `roger05` : [Fe/H]=-0.5.
– `roger10a` : [Fe/H]=-1.0, enhanced.

Description : D'origine externe, ces routines ne sont pas décrites.

Appel : Ces routines appelées par la routine générique `tdetau`, *cf.* § 8.127 (Page 207), utilisent les mêmes arguments.

8.119 Routine `rq_reac`

La fonction de cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est la gestion du calcul, en fonction de la température, de la densité et des abondances, des taux des réactions nucléaires utilisées et des énergies libérées, ainsi que de leurs dérivées par rapport à la température, à la densité et aux abondances. Suivant la valeur de la variable logique `mitler=.FALSE.` ou `mitler=.TRUE.` de la NAMELIST `NLCHIM` du fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22), l'écrantage est calculé soit par la théorie standard de l'écrantage faible, soit par celle, plus universelle, de l'écrantage de Mitler [1977] qui recouvre l'écrantage faible et intermédiaire[4]. Suivant les réactions utilisées, *cf.* § 8.76 (Page 179), la routine utilise les tabulations des taux de réactions nucléaires gérées par la routine `tabul_nuc`, *cf.* § 8.124 (Page 205).

[4]Bien que prévu, l'écrantage fort n'est pas opérant dans la version actuelle.

Description :

- Initialisation : lecture du fichier de réactions nucléaires, écritures et initialisations diverses.
- Calcul du nombre d'électrons libres, du facteur d'écran et des dérivées.
- Calcul des taux et des énergies libérées. C'est dans la tabulation des réactions nucléaires, *cf.* § 8.126 (Page 206), que sont introduits les ***dénominateurs 2 ! ou 3 !*** et les pertes par neutrinos.

Appel :

```
SUBROUTINE rq_reac(comp,t,ro,r,drt,dro,drx,q,dqt,dqo,dqx,mue,dmuex)
```

- Entrées :
  - `comp, t, ro` : Composition chimique par volume, température, densité.
- Sorties :
  - `r, drt, dro, drx, q, dqt, dqo, dqx, mue, dmuex` : Taux, énergie des réactions et dérivées par rapport à la température, densité, composition chimique, nombre d'électrons e^- par volume, et dérivées par rapport à la composition chimique.

8.120 Routine `saha`

Cette routine PUBLIC du module `mod_etat`, *cf.* § E.5 (Page 259), résout l'équation de Saha (Cox & Guili [1968], formule 15-30) pour la détermination des taux d'ionisation des divers éléments chimiques. On limite les fonctions de partition aux poids statistiques des niveaux fondamentaux. La formulation est décrite au § 7.8.4 (Page 101). Les équations, écrites sous la forme :

$$\rho N_0 \sum_i \bar{z}_i(n_e)x_i - n_e = 0$$

sont résolues par itération Newton-Raphson.

Description :

- Initialisations au premier appel :
  - On admet que les taux d'ionisation sont identiques pour tous les isotopes d'une même espèce chimique. Les différents isotopes d'une même espèce chimique sont regroupés dans un seul élément auquel l'abondance totale de l'espèce est affectée. CESAM2k signale si un des éléments utilisés est inconnu, le déroulement du calcul est alors arrêté.
  - Initialisation des potentiels d'ionisation et des poids statistiques pour les espèces identifiées par leur charge. La fonction de Fermi-Dirac $F_{1/2}$ est ensuite tabulée en fonction du paramètre de dégénérescence.
- Pour les appels suivants :
  - Initialisation des abondances de chaque espèce et du nombre d'électrons libres.
  - Calcul du facteur de dégénérescence et résolution des équations de Saha pour chaque ion, redétermination du nombre d'électrons libres. Itération Newton-Raphson sur le nombre d'électrons libres.

  - Détermination de la charge moyenne de chaque ion.
  - Après convergence : Restitution des taux d'ionisation et de la charge moyenne pour chaque isotope.

Appel : `saha` est appelée par `cesam, coeff_rota, diffm_br`.

```
SUBROUTINE saha(xchim,t,ro,ioni,z_bar,nel,eta)
```

- Entrées :
  - `xchim, t, ro` : composition chimique par volume, température, densité ;
- Sorties :
  - `ioni, z_bar, nel, eta` : taux d'ionisation et charge moyenne de chaque ion, nombre d'électrons libres, facteur de dégénérescence.

8.121 Routine `sortie`

La routine PUBLIC `sortie` du module `mod_variables`, *cf.* § E.4 (Page 255), est sans argument. Elle est appelée lorsque CESAM2k détecte une anomalie fatale au bon déroulement du calcul et à laquelle il ne peut remédier, par exemple une sortie de table. `sortie` ferme les unités d'écriture et de dessin avant l'arrêt des calculs.

8.122 Routine générique `static`

Cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), constitue la ***routine générique*** du calcul des coefficients des équations de l'équilibre quasi-statique dans l'espace physique.

Description :
`static` oriente le calcul des coefficients des équations d'équilibre quasi-statique vers leurs formes lagrangienne ou eulérienne, *cf.* § 7.2.4 (Page 71).

Appel :

```
SUBROUTINE static(fait,cx,li,y,be,ae,compt,dt,reprend,ip)
```

- Entrées :
  - `fait=1` : Calcul des résidus des équations au point de collocation.
  - `fait=2` : Calcul du résidu pour la condition au point limite.
  - `cx` : Indice du point de collocation.
  - `li` : Numéro de la limite.
  - `y` : variables et dérivées au point de collocation ou limite.
  - `compt` : Compteur du nombre d'itérations Newton-Raphson globales.
  - `dt` : Pas temporel.
  - `ip` : Indice du facteur de répartition.
- Sorties :
  - `be` : Résidus.
  - `ae` : Eléments du jacobien.
  - `reprend=.TRUE.` : La variation relative d'énergie graviphique est trop forte, le calcul devra être réinitialisé.

8.123 Routines `static_m`, `static_r`

Ces deux routines PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), sont équivalentes, elles forment les coefficients des équations de l'équilibre quasi-statique, sous leur forme lagrangienne et eulérienne respectivement, *cf.* § E.12 (Page 263). Pour chaque point de collocation, les équations de la structure interne sont écrites sous la forme $f(x) = 0$, ainsi que les conditions limites. On utilise les variables retenues pour l'intégration numérique. Ces équations ont été détaillées § 7.2 (Page 66). Bien que, formellement, la variable indépendante soit la masse (lagrangien), dans `static_m` et le rayon (eulérien) dans `static_r`, les équations sont écrites pour "l'indice réel de couche" `q`, *cf.* § 7.2 (Page 66).
Les singularités centrales des équations de la structure interne écrites sous la forme Eq. 7.3 (Page 66) disparaissent sous la forme Eq. 7.5 (Page 71) utilisée par CESAM2k. Par ailleurs, le centre n'étant pas un point de collocation, il n'y a pas lieu d'introduire explicitement les formes limites des équations. Au centre, les variables ont pour valeurs limites celles, ***sans singularité***, des polynômes par morceaux qui approchent la solution, ce qui constitue un des avantages de la méthode.
Pour les limites externes, les fonctions qui dépendent de la physique utilisée : $P_{\rm b}(L,R)$, $P_{\rm gaz\ b}(L,R)$, $T_{\rm b}(L,R)$ et $M_{\rm b}(L,R)$ sont calculées, au point $q = n$ par une des routines de physique de type `lim_atm`, *cf.* § 8.67 (Page 173).

Description :
Après quelques initialisations, on détermine la fonction de répartition et ses dérivées, puis on forme les équations et leurs dérivées par rapport à toutes les variables dépendantes. Les équations pour les conditions limites sont écrites à la fin de la routine. On a conservé dans la source des instructions permettant des vérifications, en particulier celles des coefficients du jacobien.

- Initialisations de quantités invariantes.
- Pour chaque point de collocation :
  - Extraction des variables physiques et vérification que le rayon, la masse et la luminosité[5], ont des valeurs positives.
  - Extraction de la composition chimique locale, initialisation de quantités liées à la rotation, à la pression turbulente.
  - Calcul par un appel à `thermo`, *cf.* § 8.128 (Page 207), des valeurs locales de grandeurs thermodynamiques, de l'opacité, de l'énergie nucléaire, des gradients et de leurs dérivées, transformations de ces grandeurs des unités cgs. en unités locales avec des aménagements algorithmiques.
  - Calcul de l'énergie gravifique :
    - Pour un nouveau pas temporel, à la première itération globale gérée par `resout`, *cf.* § 8.114 (Page 195), *i.e.* `new=.TRUE.`, la valeur est interpolée, en masse, à partir de ses valeurs au pas temporel précédent.
    - Pour les itérations suivantes, cette quantité est calculée avec l'alternative de l'approximation de Kippenhahn *cf.* § 7.4.1 (Page 83) ou du calcul complet *cf.* § 7.4.2 (Page 83). Au cas où, à la même abscisse lagrangienne, un écart relatif trop important de pression ou de température est observé avec le pas temporel précédent cf, § 6.2 (Page 52), la variable logique

[5] Pour `static_m`

  reprend=.TRUE. et les calculs sont réinitialisés par un retour à resout.
  - Après quelques dispositions algorithmiques concernant le moment cinétique, détermination de la fonction de répartition et des dérivées, puis formulation des équations et de leurs dérivées partielles par rapport à chacune des variables.
  - Ensemble d'instructions permettant de tester numériquement l'exactitude des dérivées.
- Pour chaque point limite :
  - Ecriture des conditions limites au centre *i.e.* li=1, 2, 3.
  - Ecriture des conditions limites à l'extérieur *i.e.* li=4, 5, 6. Pour limiter le nombre de calculs de l'atmosphère, lim_atm, *cf.* § 8.67 (Page 173), n'est appelée qu'une seule fois, les grandeurs alors calculées étant utilisées pour formuler les autres conditions externes. Il suffit d'enlever les c de commentaires pour rendre effectifs des tests de vérification des dérivées.

Appel : Ces routines appelées par la routine générique static, *cf.* § 8.122 (Page 203), utilisent les mêmes arguments.

8.124 Routine tabul_nuc

La fonction de cette routine PRIVATE du module mod_nuc, *cf.* § E.9 (Page 261), est la gestion de la tabulation des réactions nucléaires.

Description :

- Pour chaque réseau de réactions nucléaires :
  - Initialisation du nombre de réactions, du nombre d'isotopes utilisés par les réactions, des indices de ^{4}He et de la réaction 3α.
  - Initialisation des indices des isotopes utilisés dans les réactions.
  - Définition de l'ordre des réactions et du domaine de tabulation.
- Sélection du type de compilation à utiliser.
- Appel à taux_nuc, *cf.* § 8.126 (Page 206), pour le calcul des taux de ***toutes*** les réactions implémentées.
- Extraction des taux des réactions relevant du réseau retenu et tabulation.

Appel : cette routine d'initialisation tabul_nuc est appelée par rq_reac, *cf.* § 8.119 (Page 201), en début de calcul. Cette routine n'a que des arguments de sortie.

```
SUBROUTINE tabul_nuc(ar,i3al,knot_temp,m_temp,
1 nom_reac,n_temp,q0,taux_reac,temp,ttemp)
```

- nom_reac, i3al : Noms des réactions du réseau utilisé et indice de la réaction 3α.
- ar, q0, taux_reac : Tabulation des masses réduites, des énergies, des taux de réaction.
- knot_temp, m_temp, n_temp, temp, ttemp : Eléments du vecteur nodal de la tabulation des taux de réaction.

8.125 Routine `taueff`

Cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), a pour fonction la détermination de la profondeur optique $\tau_\star$ correspondant au rayon bolométrique du modèle $R_\star$. $\tau_\star$ vérifie l'équation implicite :

$$T_{\rm eff} = \mathcal{T}(\tau_\star, T_{\rm eff}, g).$$

Description :
- Initialisation $\tau_\star = 2/3$.
- Détermination de $\tau_\star$ par l'algorithme de Newton-Raphson avec appel à la routine de la loi $T(\tau)$ concernée, *cf.* § 8.127 (Page 207).

Appel : `taueff` est appelée par `lim_atm`, *cf.* § 8.67 (Page 173).

```
SUBROUTINE taueff(teff,grav,tau)
```

- Entrées :
  - `teff, grav` : température effective, gravité.
- Sorties :
  - `tau` : $\tau_\star$.

8.126 Routine `taux_nuc`

La fonction de cette routine PRIVATE du module `mod_nuc`, *cf.* § E.9 (Page 261), est l'initialisation des paramètres nucléaires, le calcul des énergies et des taux d'un grand nombre de réactions. Il est évidemment possible d'implémenter de nouvelles réactions, la procédure à suivre en est détaillée au § 5.1 (Page 49).

Description :
Après une initialisation, les taux de réactions sont dérivés des formules de diverses compilations. C'est la nécessité du calcul des dérivées qui a motivé la tabulation des taux de réactions nucléaires plutôt que l'emploi direct des formules.
- Au premier appel :
  - Initialisations diverses.
  - Initialisation des excès de masse.
  - Initialisation des masses atomiques.
  - Initialisation des charges des isotopes.
  - Initialisation des noms des isotopes.
  - Initialisation des noms des réactions, des masses réduites, des énergies, des charges des noyaux de la réaction.
- Initialisations diverses reliées à la température.
- Calcul des taux de ***toutes*** les réactions, suivant la compilation retenue.

Appel : `taux_nuc` est appelée par `tabul_nuc`, *cf.* § 8.124 (Page 205).

```
SUBROUTINE taux_nuc(t,total,rt,zit,izzt,qt,nom_react,nucleot,
1 at,nom_elemt)
```

- Entrées :

  - `t, total` : température, indice du choix de la compilation à utiliser, *cf.* § 3.3 (Page 22).
- Sorties :
  - `rt` : ln(taux des réactions).
  - `zit` : Charges des noyaux.
  - `izzt` : Charges des noyaux de la réaction.
  - `qt` : Energies.
  - `nom_react` : Symboles des réactions.
  - `nucleot` : Masses des isotopes .
  - `at` : Masses réduites.
  - `nom_elemt` : Noms des isotopes.

8.127 Routine générique `tdetau`

Cette routine PUBLIC du module `mod_atm`, *cf.* § E.8 (Page 260), constitue la routine générique de la gestion du calcul de la température en fonction de la profondeur optique.

Description :
Le calcul est orienté vers la routine de loi $T(\tau)$ indiquée dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22).

Appel : Les lois $T(\tau)$ étant utilisées pour reconstruire une atmosphère, la routine `tdetau` est appelée de `lim_atm`, *cf.* § 8.67 (Page 173).

```
SUBROUTINE tdetau(tau,teff,grav,t,dtsdtau,dtsdteff,dtsdg,
1 ro_ext,dro_grav,dro_teff,f_tau,df_tau,d2f_tau)
```

- Entrées :
  - `tau, teff, grav` : profondeur optique, température effective, gravité.
- Sorties :
  - `t, dtsdtau, dtsdteff, dtsdg` : température et dérivées ;
  - `ro_ext, dro_grav, dro_teff` : densité externe et dérivées ;
  - `f_tau, df_tau, d2f_tau` : fonction f et dérivées *cf.* § 7.3.3 (Page 78).

8.128 Routine `thermo`

Cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), effectue le calcul des variables thermodynamiques au sens large, à l'aide d'appels aux diverses routines génériques.

Description
- Initialisations effectuées lors du premier appel.
- Appel à l'équation d'état pour le calcul des grandeurs thermodynamiques.
- Détermination de l'énergie thermonucléaire.
- Détermination de l'opacité.
- Calcul du gradient radiatif.
- Formation du critère de convection (Schwarzschild ou Ledoux).

- Affectation du gradient adiabatique ou radiatif dans les extensions des zone convective, *cf.* § 3.8 (Page 29).
- Pour les zones convectives, calcul du gradient.

Appel : `thermo` est appelée de divers endroits en particulier par `lim_zc`, *cf.* § 8.69 (Page 174), `static_m`, `static_r`, *cf.* § 8.123 (Page 204).

```
SUBROUTINE thermo(pt,p,t,m,l,r,dlpp,xchim,dxchim,
1 ro,drop,drot,drox,u,dup,dut,dux,grad,dgradpt,dgradp,dgradt,
2 dgradx,dgradm,dgradl,dgradr,dgradlpp,
3 gam,dgampt,dgamp,dgamt,dgamx,dgamm,dgaml,dgamr,dgamlpp,
4 epsilon,depsp,depst,depsx,kap,dkapp,dkapt,dkapx,
5 delta,deltap,deltat,deltax,cp,dcpp,dcpt,dcpx,
6 gradad,dgradadp,dgradadt,dgradadx,
7 hp,dhppt,dhpp,dhpt,dhpx,dhpr,dhpm,
8 gradrad,alfa,beta,gamma1,radiatif)
```

- Entrées :
  - `pt, p, t, m, l, r` : Pression totale, pression gazeuse, température, masse, luminosité, rayon.
  - `xchim, dxchim, dlpp` : Composition chimique par mole, dérivée / $(M/M_\odot)$, $\frac{d\ln P_{\rm gaz}}{d\ln P_{\rm tot}}$.
- Sorties :
  - `ro, drop, drot, drox` : densité et dérivées ;
  - `u, dup, dut, dux` : énergie interne et dérivées ;
  - `grad, dgradpt, dgradp, dgradt, dgradx, dgradm, dgradl, dgradr, dgradlpp` : gradient de température $\nabla \equiv \frac{\partial \ln T}{\partial \ln P_{\rm tot}}$ et dérivées ;
  - `gam, dgampt, dgamp, dgamt, dgamx, dgamm, dgaml, dgamr, dgamlpp` : efficacité de la convection et dérivées ;
  - `epsilon, depsp, depst, depsx` : énergie nucléaire et gravifique et dérivées ;
  - `kap, dkapp, dkapt, dkapx` : opacité et dérivées ;
  - `delta, deltap, deltat, deltax` : δ et dérivées ;
  - `cp, dcpp, dcpt, dcpx` : $c_{\rm p}$ et dérivées ;
  - `gradad, dgradadp, dgradadt, dgradadx` : gradient adiabatique et dérivées ;
  - `hp, dhppt, dhpp, dhpt, dhpx, dhpr, dhpm` : échelle de hauteur de pression et dérivées ;
  - `gradrad, alfa, beta, gamma1, radiatif` : gradient radiatif, α, β, Γ_1, `radiatif`=.TRUE. : on est dans une zone radiative.

8.129 Routine `thermo_atm`

Pour la restitution de l'atmosphère, cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), est l'homologue de la précédente. Elle n'en diffère que par :

- l'abscence d'appel aux taux des réactions thermonucléaires,
- le calcul du gradient de température qui tient compte de la loi $T(\tau)$ utilisée pour la restitution de l'atmosphère *cf.* § 7.3.2 (Page 76),

– la composition chimique, la luminosité et la vitesse angulaire y sont spatialement constantes.

Description :
– Initialisations effectuées lors du premier appel.
– Appel à l'équation d'état pour le calcul des grandeurs thermodynamiques.
– Détermination de l'opacité.
– Calcul du gradient radiatif.
– Formation du critère de convection (Schwarzschild ou Ledoux).
– Pour les zones convectives, calcul du gradient.

Appel : `thermo_atm` est appelée par `eq_atm`, *cf.* § 8.45 (Page 162).

```
SUBROUTINE thermo_atm(pt,p,t,xchim,m,l,r,dlpp,
1 tau,df_tau,d2f_tau,rstar,ro,drop,drot,kap,dkapp,dkapt,gradad,
2 dgradadp,dgradadt,grad,dgradpt,dgradp,dgradt,dgradr,dgradrs,
3 dgradm,dgradtau,dgradlpp,gam,dgampt,dgamp,dgamt,dgamr,dgamrs,
4 dgamm,dgamtau,dgamlpp,hp,dhppt,dhpp,dhpt,dhpr,dhpm,delta,deltap,
5 deltat,cp,gradrad,alfa,beta,gamma1,radiatif,deriv)
```

– Entrées :
  – `pt, p, t, m, l, r` : pression totale, pression gazeuse, température, masse, luminosité, rayon,
  – `xchim, dlpp` : composition chimique par mole, $\frac{d\ln P_{\rm gaz}}{d\ln P_{\rm tot}}$;
  – `tau, df_tau, d2f_tau, rstar` : profondeur optique, partie dépendante de τ de la loi $T(\tau)$ et dérivées premières et secondes, rayon total ;
  – `deriv`=.TRUE. : le calcul des dérivées est requis.
– Sorties :
  – `ro, drop, drot` : densité et dérivées ;
  – `grad, dgradpt, dgradp, dgradt, dgradr, dgradrs, dgradm, dgradtau, dgradr, dgradlpp` : gradient de température $\nabla \equiv \frac{\partial\ln T}{\partial\ln P_{\rm tot}}$ et dérivées ;
  – `gam, dgampt, dgamp, dgamt, dgamr, dgamrs, dgamm, dgamtau, dgamlpp` : efficacité de la convection et dérivées ;
  – `kap, dkapp, dkapt` : opacité et dérivées ;
  – `gradad, dgradadp, dgradadt` : gradient adiabatique et dérivées ;
  – `hp, dhppt, dhpp, dhpt, dhpr, dhpm` : échelle de hauteur de pression et dérivées ;
  – `delta, deltap, deltat` : δ et dérivées ;
  – `cp, gradrad, alfa, beta, gamma1, radiatif` : $c_{\rm p}$, gradient radiatif, α, β, Γ_1, `radiatif`=.TRUE. : on est dans une zone radiative.

8.130 Routine `trho`

La fonction de cette routine PRIVATE du module `mod_atm`, *cf.* § E.8 (Page 260), est la lecture des données et l'interpolation en (T,ρ) des lois $T(\tau)$ de type `roger**`, *cf.* § 8.118 (Page 201).

Description : D'origine externe cette routine n'est pas décrite.

Appel : `trho` est appelée des routines de lois $T(\tau)$ du type `roger**`.

8.131 Routine `update`

Suivant la valeur `.TRUE. / .FALSE.` de la variable logique `next`, cette routine PRIVATE du module `mod_static`, *cf.* § E.12 (Page 263), permet, soit de translater au temps t le modèle calculé pour le temps $t + dt$, soit de réinitialiser un modèle pour le temps $t + dt$ avec celui obtenu pour le temps t. L'extension `_t` permet de différencier les variables au temps t de celles au temps $t + dt$.

Description :

- Si `next=.TRUE.` les grandeurs caractérisant le modèle au temps $t + dt$ sont transcrites sur celles du modèle au temps t, le pas temporel est ajusté de façon à éventuellement arrêter le calcul sur l'âge maximum désiré.
- Si `next=.FALSE.` les grandeurs caractérisant le modèle au temps t sont transcrites sur celles du modèle au temps $t + dt$, le pas temporel est ajusté de façon à éventuellement arrêter le calcul sur l'âge maximum désiré.

Appel : `update` est appelée par `resout`, *cf.* § 8.114 (Page 195).

```
SUBROUTINE update(next,dt,dts)
```

- Entrées :
  - `next=.TRUE.` : On écrase les variables sur celles du pas temporel précédent.
  - `next=.FALSE.` : On restitue les variables du pas temporel précédent.
- Entrées/Sorties :
  - `dt, dts` : Pas temporel, pas temporel estimé.

8.132 Fonction `u_initial`

Cette fonction PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), initialise la vitesse de circulation méridienne U du formalisme de la diffusion du moment cinétique, *cf.* § 7.9.1 (Page 111).

Description : La valeur initiale de la vitesse de circulation méridienne est initialisée par :

$$|\Omega| U_0 \left(2 - \frac{R}{R_\star)^2}\right)^{-1}, \quad U_0 \sim 0.01.$$

Appel : `u_initial` est appelée par `initialise_rota`, *cf.* § 8.62 (Page 171).

```
FUNCTION u_initial(r,rstar)
```

- `r, rstar` : rayon local, rayon total.

8.133 Routine `vent`

Lorsqu'il y a perte ou gain de masse, CESAM2k offre la possibilité que la composition chimique de la matière gagnée ou perdue diffère de celle des couches les plus externes *cf.* § 7.8.1 (Page 98). Cette possibilité est gérée par la routine PUBLIC `vent` du module `mod_nuc`, *cf.* § E.9 (Page 261).

Description :

- Lors d'un appel d'initialisation, depuis la routine `evol`, on effectue une recherche dans l'environnement du fichier où sont indiquées les fractions de masse du vent, *cf.* § 4.5 (Page 40). Si un tel fichier est trouvé, les fractions de masse sont normalisées, puis réparties entre les différents isotopes du réseau nucléaire, et enfin transformées en fraction par mole ; la variable logique `l_vent` est déclarée `.TRUE.` Si le fichier est inconnu, la variable logique `l_vent` est déclarée `.FALSE.`, la composition chimique du vent sera celle de la couche la plus externe du modèle.
- Lors des appels suivants, une correction aux taux de réactions nucléaires est effectuée, *cf.* § 7.8.1 (Page 98).

Appel : `vent` est appelée par :
- `evol`, *cf.* § 8.55 (Page 167), pour initialisation et dans le cas où le vent a une composition chimique différente de celle de la couche la plus externe.
- `rk_imps`, *cf.* § 8.117 (Page 199) ou `eq_dif_chim`, *cf.* § 8.38 (Page 159), à la suite d'un appel à la routine `nuc`, *cf.* § 8.76 (Page 179).

Les arguments sont optionnels.

```
SUBROUTINE vent(xchim,dxchim,jac)
```

- Entrées :
  - `xchim` : Composition chimique par mole.
- Entrées/Sorties :
  - `dxchim, jac` : Taux de réactions nucléaires et jacobien.

8.134 Routine `write_nl`

Cette routine PUBLIC du module `mod_exploit`, *cf.* § E.14 (Page 264), a pour fonction la formation d'un fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22).

Description : Les NAMELISTs sont écrites après l'ouverture du fichier.

Appel : `write_nl`, qui n'a pas d'argument, est utilisée par des programmes d'exploitation, *cf.* § 3.17 (Page 35).

8.135 Fonction `w_initial`

Cette fonction PRIVATE du module `mod_cesam`, *cf.* § E.13 (Page 264), initialise la vitesse angulaire Ω du formalisme de la diffusion du moment cinétique, *cf.* § 7.9.1 (Page 111).

Description : La valeur locale de la vitesse angulaire est donnée par :

$$\Omega = |\Omega_0| \left\{1 + \Omega_\mathrm{f}\left[1 + \cos\left(\pi \frac{R}{R_\star}\right)\right]\right\},$$

Ω_0 et Ω_f sont respectivement la vitesse angulaire initiale et le facteur de forme définis dans le fichier de données `mon_modele.don`, *cf.* § 3.3 (Page 22).

Appel : `w_initial` est appelée par `initialise_rota`, *cf.* § 8.62 (Page 171).

```
FUNCTION theta_initial(r,rstar)
```

- `r, rstar` : rayon local, rayon total.

8.136 Package z14xcotrin21

Le package d'interpolation d'opacité `z14xcotrin21` de A.I.Boothroyd, est utilisé par CESAM2k pour le calcul de l'opacité, lorsque l'abondance en éléments lourds dépasse $Z > 0.1$. Les routines d'interpolations utilisent des tables du sous-directory `SUN_STAR_DATA`, *cf.* § 1.3 (Page 2). Elles nécessitent une mise en forme qui est transparente pour l'utilisateur *cf.* § 2.2.2 (Page 14).
La mise en œuvre du package `z14xcotrin21` a bénéficié d'une collaboration avec A.I.Boothroyd. En raison de la complexité de sa programmation, `z14xcotrin21` n'est pas inclus dans un module. Le package est compilé conjointement au module `mod_opa`, *cf.* § E.6 (Page 259).

Description :
En raison de leur complexité et de leur origine externe, les routines du package ne sont pas détaillées. Pour plus d'informations se reporter aux explications données en tête de la source de `z14xcotrin21`.

Appel :
Avec le package `z14xcotrin21` CESAM2k utilise la routine `opal_x_cno_fu` ou à défaut la routine `opal`. Ces routines sont appelées par `opa_opal2`, *cf.* § 8.81 (Page 184).
En raison de la complexité, les listes d'appel ne sont pas détaillées.

Chapitre 9

Les routines numériques

Ces routines sont des routines PUBLIC ou PRIVATE du module `mod_numerique`, *cf.* § E.2 (Page 247) ou encore `mod_exploit`, *cf.* § E.14 (Page 264). Elles relèvent de l'analyse numérique et de l'algorithmique. Certaines routines sont originales, d'autres ont été construites en s'inspirant des descriptions données dans des ouvrages d'analyse numérique élémentaire. On n'en donne qu'une description sommaire. Ces routines sont regroupées dans le sous-directory `SOURCE`.

9.1 Routines numériques et assimilées

9.1.1 Routine `arb_rom`

Cette fonction transforme l'entier $i \in [0, 30]$ en notation arabe en notation romaine, Exemple : 21 $\Longrightarrow XXI$. La notation romaine comporte au plus 10 caractères, au delà, la sortie est CCC. Cette fonction est utilisée pour noter les niveaux d'ionisation.

9.1.2 Routine `boite`

Cette routine permet de dessiner une boîte d'erreur dont on donne les coordonnées du centre et la longueur des cotés.

9.1.3 Routine `box`

Cette routine permet de dessiner une boîte de dimensions $\pm\Delta x \times \Delta y$ autour du point de coordonnées (x, y).

9.1.4 Routine `delete_doubles`

Cette routine permet de supprimer les éléments succéssifs identiques dans un tableau ALLOCATABLE ordonné par ordre croissant ; si besoin la dimension du tableau est réajustée.

9.1.5 Routine `difdiv`

Calcul des différences divisées pour interpolation par la formule de Newton *cf.* , Stoer & Bulirsch [1979].

9.1.6 Routine `fermi_dirac`

Approximations analytiques des intégrales de Fermi-Dirac, reprises du package MHD, *cf.* § 8.53 (Page 166).

9.1.7 Routine `gauss_band`

Routine de résolution d'un système linéaire bande, dont la largeur de la bande est constante et comportant plusieurs seconds membres. L'algorithme est celui de l'élimination de Gauss avec équilibrage et pivot partiel. L'espace mémoire utilisé correspond à la partie non identiquement nulle du système.

9.1.8 Routine `horner`

La subroutine `horner` permet de calculer, pour $x = \alpha$, les dérivées de tous ordres d'un polynôme ainsi que le quotient de ce polynôme par $x - \alpha$ (algorithme de Hörner).

9.1.9 Routine `intgauss`

Initialisation des poids et des abscisses pour l'intégration de Gauss de divers ordres.

9.1.10 Routine `matinv`

Programme d'inversion de matrice. Utilise l'élimination de Gauss avec pivot total de `gauss_band`.

9.1.11 Routine `min_max`

Recherche du maximum et du minimum d'une table. Cette routine est adaptée au calcul des échelles pour des tracés.

9.1.12 Routine `min_max_cond`

Recherche du maximum et du minimum d'une table sous condition. Cette routine est adaptée au calcul des échelles pour des tracés.

9.1.13 Routine `neville`

Algorithme de Neville pour interpolation polynômiale, *cf.* , Stoer & Bulirsch [1979].

9.1.14 Routine `newton`

Interpolation polynômiale avec dérivées, par la formule de Newton, *cf.* , Conte & de Boor [1987].

9.1.15 Routine `pause`

Pause avec commentaire et poursuite/arrêt.

9.1.16 Routine `polyder`

C'est une autre version de l'algorithme de Hörner dont les arguments diffèrent de ceux de `horner`. Pour des raisons historiques CESAM2k fait appel aux deux versions.

9.1.17 Routine `shell`

Routine tri d'un tableau optimisé pour F95.

9.1.18 Routine `zoning`

Permet de déterminer la répartition des abscisses de façon à ce que les incréments des ordonnées soient constants.

9.2 Routines spécifiques aux B-splines

Ces routines sont originales dans leur grande majorité. Les algorithmes utilisés sont, pour la plupart, inspirés de ceux décrits par C. de Boor [1978] et Schumaker [1981].

9.2.1 Routine `bsp1ddn`

Généralise `bsp1dn` en calculant toutes les dérivées non identiquement nulles. Ne doit être utilisée que si des dérivées d'ordre supérieur à l'unité sont requises car elle demande plus de ressources que `bsp1dn`.

9.2.2 Routine `bsp1dn`

Interpolation au même point de `n` fonctions développées sur une base de B-splines ; calcul du développement si nécessaire.

9.2.3 Routine `bsp_dis`

Routine de calcul des coefficients des B-splines pour `n` fonctions avec discontinuités pour interpolations ou lissage par Bézier.

9.2.4 Routine `bval0`

Routine qui calcule la valeur de toutes les B-splines non identiquement nulles en un point fixé, *cf.* § 7.1.1 (Page 60).

9.2.5 Routine `bval1`

Routine qui calcule la valeur de toutes les B-splines non identiquement nulles et leurs dérivées premières en un point fixé. Demande plus de ressources que `bval0`.

9.2.6 Routine `bvald`

Routine qui calcule la valeur de toutes les B-splines non identiquement nulles ainsi que toutes leurs dérivées en un point fixé. Demande plus de ressources que `bval1`.

9.2.7 Routine `coll`

Routine qui génère les points de collocation de la base de C. de Boor [1978] pour l'intégration des équations différentielles.

9.2.8 Fonction `colpnt`

Fonction calculant l'abscisse du i-ième point de collocation dans l'intervalle $]x(l), x(l+1)[$ pour la résolution d'une équation différentielle d'ordre r. `colpnt` est appelée par `coll`.

9.2.9 Routine `linf`

Localisation d'un point dans une table d'abscisses en ordre strictement croissant, adapté aux B-splines.

9.2.10 Routine `newspl`

Effectue un changement de base de B-splines pour `n` polynômes par morceaux.

9.2.11 Routine `noedif`

Routine d'initialisation du réseau de points de table de la base de C. de Boor [1978], pour la résolution des équations différentielles par collocation.

9.2.12 Routine `noein`

Détermine la séquence de nœuds de raccord pour une interpolation "optimale" suivant C. de Boor [1978].

9.2.13 Routine `noeu_dis`

Détermine la séquence de nœuds de raccord pour une interpolation "optimale" en tenant compte de discontinuités ; identique a `noein` s'il n'y a pas de discontinuité.

9.2.14 Routine `noeud`

Détermine la séquence de nœuds de raccord pour un vecteur des multiplicités donné.

9.2.15 Routine `schu58_n`

Calcul des valeurs en un point de `n` polynômes par morceaux à partir de leurs développements sur une base de B-splines et des valeurs des B-splines non identiquement nulles en ce point, *cf.* Schumaker [1981].

9.2.16 Routine `sum_n`

Calcule les intégrales de `n` splines entre des bornes données. Les coefficients doivent avoir été précédemment calculés par `bsp1dn` par exemple. Au premier appel, ces coefficients seront adaptés au calcul de l'intégrale, donc modifiés. Ils ne seront pas recalculés lors des appels ultérieurs. S'inspire de l'algorithme 5.19 de Schumaker [1981].

Troisième partie

APPENDICES

Annexe A

Installation sous LINUX

A.1 Création de la bibliothèque avec `ifc`

La Norme F95, telle qu'elle a été définie en 1996, est implémentée de façon stricte dans le compilateur[1] `ifc`. Depuis, on s'est aperçu d'une limitation "idiote" de cette Norme, interdisant de mettre un tableau `ALLOCATABLE` alors que :

`REAL(kind=dp), INTENT(out), POINTER, DIMENSION( :)`

était autorisé. Dans certains contextes, `POINTER` et `ALLOCATABLE` sont synonymes. Quelques temps plus tard, un Technical Report qui "corrigeait" et autorisait cette limitation est paru. Les compilateurs de Compaq/Digital (stations Unix, PC Windows) Lahey-Fujitsu (Linux, Windows) ont été corrigés ; INTEL semble n'avoir effectué la correction que dans la dernière version de `ifc`.

Remède : Remplacer `ALLOCATABLE` par `POINTER`, chaque fois que le compilateur manifeste. Par la suite, avec le compilateur `ifc` à jour, tout rentrera dans l'ordre. Actuellement la Norme Fortran 2003 différencie `ALLOCATABLE` et `POINTER`.

Les corrections à apporter sont les suivantes :

1. Routine `tabul_nuc.f` transformer :

   ```
   REAL (kind=dp), INTENT(out), ALLOCATABLE, DIMENSION(:,:) :: taux_reac
   REAL (kind=dp), INTENT(out), ALLOCATABLE, DIMENSION(:) ::ar,q0,temp,ttemp
   INTEGER, INTENT(out) :: i3al, knot_temp, m_temp, n_temp
   CHARACTER (len=20), INTENT(out), ALLOCATABLE, DIMENSION(:) :: nom_reac
   ```

   en :

   ```
   REAL (kind=dp), POINTER, DIMENSION(:,:) :: taux_reac
   REAL (kind=dp), POINTER, DIMENSION(:):: ar, q0, temp, ttemp
   INTEGER, INTENT(out) :: i3al, knot_temp, m_temp, n_temp
   CHARACTER (len=20), POINTER, DIMENSION(:) :: nom_reac
   ```

2. Routine `tabul_nuc`, transformer :

   ```
   REAL (kind=dp), ALLOCATABLE, SAVE, DIMENSION(:,:) :: decrx,dzetax,taux_reac
   REAL (kind=dp), ALLOCATABLE, SAVE, DIMENSION(:) :: ar, datx, dbidx,
   1 dc1x, decro, decrt, dfx, df12x, dkx, dnex, dr, dzstarx, ecran,
   2 lambda, mz1z2, q0, temp, ttemp, tx, z1z2
   ```

[1] Diagnostique, commentaires et remèdes selon B.Pichon.

```
REAL (kind=dp), SAVE, DIMENSION(2) :: zeta, dzetat, dzetaro
REAL (kind=dp), SAVE :: cte1, cte2, cte3, cte4, cte5, cte6, cte7,
1 cte8, cte9, cte10
REAL (kind=dp) :: zstar, ne, c1, f12, at, k, bid, bid1, df,
1 e0, eps, beta, dnero, dc1ro, df12t, df12ro, datt, datro, dkt,
2 dkro, dbidt, dbidro, dft, dfro
INTEGER, SAVE :: i3al, knot_temp, l=1, m_temp, n_temp
INTEGER :: i, j
CHARACTER (len=20), ALLOCATABLE, DIMENSION(:) :: nom_reac
```

en :

```
REAL (kind=dp), POINTER, SAVE, DIMENSION(:,:) :: decrx,dzetax,taux_reac
REAL (kind=dp), POINTER, SAVE, DIMENSION(:) :: ar, datx, dbidx,
1 dc1x, decro, decrt, dfx, df12x, dkx, dnex, dr, dzstarx, ecran,
2 lambda, mz1z2, q0, temp, ttemp, tx, z1z2
REAL (kind=dp), SAVE, DIMENSION(2) :: zeta, dzetat, dzetaro
REAL (kind=dp), SAVE :: cte1, cte2, cte3, cte4, cte5, cte6, cte7,
1 cte8, cte9, cte10
REAL (kind=dp) :: zstar, ne, c1, f12, at, k, bid, bid1, df,
1 e0, eps, beta, dnero, dc1ro, df12t, df12ro, datt, datro, dkt,
2 dkro, dbidt, dbidro, dft, dfro
INTEGER, SAVE :: i3al, knot_temp, l=1, m_temp, n_temp
INTEGER :: i, j
CHARACTER (len=20), POINTER, DIMENSION(:) :: nom_reac
```

3. Routine, `mod_donnees` transformer :

```
CHARACTER (len=4), SAVE, PUBLIC, ALLOCATABLE, DIMENSION(:) :: nom_elem
```

en :

```
CHARACTER (len=4), SAVE, PUBLIC, POINTER, DIMENSION(:) :: nom_elem
```

A.2 Installation de PGPLOT

A.2.1 Installation intégrale avec `ifc`

Thierry Corbard, Observatoire de la Côte d'Azur, a construit le fichier de configuration `ifc_gcc.conf` de PGPLOT avec LINUX pour le compilateur F90 `ifc` de INTEL. Pour créer la source de PGPLOT en utilisant ce compilateur :

1. Se placer dans le sous-directory `/usr/local/src/pgplot/sys_linux` de la source de PGPLOT.
   (a) Y inclure le fichier de configuration : `ifc_gcc.conf`, que l'on trouvera dans le sous-directory `SCRIPTS`.
   (b) Y créer le sous-directory `ifc_src` : `mkdir ifc_src`.
   (c) Y inclure le fichier `ifc_src/grgenv.f`, que l'on trouvera dans le sous-directory `SCRIPTS`.
2. Exécuter :
   `/usr/local/src/pgplot/makemake /usr/local/src/pgplot linux ifc_gcc`

A.2.2 Installation sans larme

Sous LINUX, il est possible d'installer PGPLOT sans trop de difficultés en procédant de la façon[2] suivante :

1. Avec un moteur de recherche, Google par exemple, rechercher `pgplot rpm`.
2. Parmi les sites détectés, choisir celui qui propose de télécharger une version récente correspondant au processeur et au système d'exploitation utilisés. Par exemple, `pgplot-5.2.2-1.i686.rpm` est satisfaisant pour un Pentium4 sous Mandrake[3].
3. Télécharger ce logiciel. A l'issue du téléchargement, se placer dans le directory où s'est effectué le téléchargement. Cliquer sur le nom du logiciel téléchargé, acquérir les droits du super-utilisateur, les logiciels s'installent.
4. Le système place les deux bibliothèques partageables `libpgplot.so.5.2.0` et `libpgplot.so.5` dans le directory `/usr/lib` et y crée le sous-directory `pgplot`. Il contient les 3 fichiers : `grexec.f, rgb.txt, grfont.dat.`
5. Créer dans le sous-directory `pgplot` deux liens vers les bibliothèques partageables : `ln -s /usr/lib/libpgplot.so.5 libpgplot.so,`
   `ln -s /usr/lib/libpgplot.so.5.0 libpgplot.so.0`
6. Dans le fichier `.cshrc` ajouter :
   (a) Le chemin de `PGPLOT LD_LIBRARY_PATH **** :/usr/lib/pgplot`.
   (b) La variable d'environnement `setenv PGPLOT_DIR /usr/lib/pgplot`.

A.3 Exemple de MAKEFILE

Le `MAKEFILE` suivant, créé par Th. Corbard O.C.A., est placé dans le sous-directory `EXPLOIT`. Les paramètres correspondent au compilateur `ifc` d'Intell et les chemin à une implantation standard. Son utilisation est décrite au § 1.6.1 (Page 5).

```
RM=rm -f

AR=ar ruv

FC90 = ifort

mod_ext=mod #extension for module files

FFLAGS90  =  -extend_source -O -xW -traceback -save -nbs

OPA_HOUDEK_PATH=../SUN_STAR_DATA/v9/lib
```

[2]Cette procédure a été testée avec les distributions LINUX Mandrake et SuSE.
[3]On trouvera ce logiciel dans le sous-directory `SCRIPTS` de la distribution.

```
LIBLINK=   -L. -lcesam2k -L$(OPA_HOUDEK_PATH) -lopint -lpgplot -lX11

#Adapt the following if you need to specify the location of X11 and pgplot lib
#PGPLOT_DIR=/usr/local/pgplot
#X11_DIR=/usr/lib64
#LIBLINK=  -L. -lcesam2k -L$(PGPLOT_DIR) -lpgplot -L$(X11_DIR) -lX11
#Mis à jour pour SOURCE et EXPLOIT du 01.03.06, P.Morel

SRCS =  \
mod_kind.f\
mod_numerique.f\
mod_donnees.f\
mod_variables.f\
mod_etat.f\
mod_opa.f\
mod_conv.f\
mod_atm.f\
mod_nuc.f\
mod_bp_for_alecian.f\
mod_evol.f\
mod_static.f\
mod_cesam.f\
mod_exploit.f

OBJS =   \
mod_kind.o\
mod_numerique.o\
mod_donnees.o\
mod_variables.o\
mod_etat.o\
mod_opa.o\
mod_conv.o\
mod_atm.o\
mod_nuc.o\
mod_bp_for_alecian.o\
mod_evol.o\
mod_static.o\
mod_cesam.o\
mod_exploit.o

MODS =    \
mod_kind.$(mod_ext)\
mod_numerique.$(mod_ext)\
mod_donnees.$(mod_ext)\
mod_variables.$(mod_ext)\
mod_etat.$(mod_ext)\
mod_opa.$(mod_ext)\
```

```
mod_conv.$(mod_ext)\
mod_atm.$(mod_ext)\
mod_nuc.$(mod_ext)\
mod_bp_for_alecian.$(mod_ext)\
mod_evol.$(mod_ext)\
mod_static.$(mod_ext)\
mod_cesam.$(mod_ext)\
mod_exploit.$(mod_ext)

cesam2k.out: cesam2k.f libcesam2k.a
$(FC90) $(FFLAGS90) -o cesam2k.out cesam2k.f $(LIBLINK)

libcesam2k.a:  $(MODS)
$(AR) libcesam2k.a $(OBJS)
$(RM) $(OBJS)

mod_kind.o mod_kind.$(mod_ext): mod_kind.f
$(FC90) $(FFLAGS90) -c $*.f

mod_numerique.o mod_numerique.$(mod_ext): \
        arb_rom.f\
        boite.f \
        box.f \
        bsp1dn.f\
        bsp1ddn.f\
        bsp_dis.f\
        bval0.f\
        bval1.f\
        bvald.f\
        coll.f\
        colpnt.f\
        difdiv.f\
        fermi_dirac.f\
        gauss_band.f\
        horner.f\
        intgauss.f\
        linf.f\
        matinv.f\
        neville.f\
        newspl.f\
        newton.f\
        noedif.f\
        noein.f\
        noeud.f\
        noeu_dis.f\
        pause.f\
        polyder.f\
```

```
        schu58_n.f\
        shell.f\
        sum_n.f\
        zoning.f\
mod_kind.$(mod_ext)\
mod_numerique.f
$(FC90) $(FFLAGS90) -c $*.f

mod_donnees.o mod_donnees.$(mod_ext): \
        ctes_85.f\
        ctes_94.f\
        ini_ctes.f\
        lit_nl.f\
        print_ctes.f\
        mod_kind.$(mod_ext)\
        mod_numerique.$(mod_ext)\
        mod_donnees.f
$(FC90) $(FFLAGS90) -c $*.f

mod_variables.o mod_variables.$(mod_ext): \
        chim_gram.f\
        inter.f\
        sortie.f\
        mod_kind.$(mod_ext)\
        mod_numerique.$(mod_ext)\
        mod_donnees.$(mod_ext)\
        mod_variables.f
$(FC90) $(FFLAGS90) -c $*.f

mod_etat.o mod_etat.$(mod_ext): \
         df_rotx.f\
         etat.f\
         etat_ceff.f\
         etat_eff.f\
         etat_gong1.f\
         etat_gong2.f\
         etat_mhd.f\
         etat_opal.f\
         etat_opalX.f\
         etat_opalZ.f\
         saha.f\
         mod_kind.$(mod_ext)\
         mod_numerique.$(mod_ext)\
         mod_donnees.$(mod_ext)\
         mod_etat.f
$(FC90) $(FFLAGS90) -c $*.f
```

```
mod_opa.o mod_opa.$(mod_ext): \
        kappa_cond.f\
        opa.f\
        opa_gong.f\
        opa_houdek9.f\
        opa_int_zsx.f\
        opa_opalCO.f\
        opa_opal2.f\
        opa_yveline.f\
        opa_yveline_lisse.f\
        z14xcotrin21.f\
        mod_kind.$(mod_ext)\
        mod_numerique.$(mod_ext)\
        mod_donnees.$(mod_ext)\
        mod_variables.$(mod_ext)\
        mod_opa.f
$(FC90) $(FFLAGS90) -c $*.f

mod_conv.o mod_conv.$(mod_ext): \
        conv.f\
        conv_a0.f\
        conv_cgm_reza.f\
        conv_cm.f\
        conv_cml.f\
        conv_cm_reza.f\
        conv_jmj.f\
        mod_kind.$(mod_ext)\
        mod_donnees.$(mod_ext)\
        mod_variables.$(mod_ext)\
        mod_conv.f
$(FC90) $(FFLAGS90) -c $*.f

mod_atm.o mod_atm.$(mod_ext): \
        atm.f\
        coll_atm.f\
        edding.f\
        eq_atm.f\
        hopf.f\
        k5750.f\
        k5777.f\
        lim_atm.f\
        lim_gong1.f\
        lim_tau1.f\
        roger00.f\
        roger02.f\
        roger05.f\
        roger10a.f\
```

```
         taueff.f\
         tdetau.f\
         thermo_atm.f\
         trho.f\
         mod_kind.$(mod_ext)\
         mod_numerique.$(mod_ext)\
         mod_donnees.$(mod_ext)\
         mod_variables.$(mod_ext)\
         mod_etat.$(mod_ext)\
         mod_opa.$(mod_ext)\
         mod_conv.$(mod_ext)\
         mod_atm.f
$(FC90) $(FFLAGS90) -c $*.f

mod_nuc.o mod_nuc.$(mod_ext): \
         abon_ini.f\
         iben.f\
         nuc.f\
 planetoides.f\
         pp1.f\
         pp3.f\
         ppcno10BeBFe.f\
         ppcno10Fe.f\
         ppcno10K.f\
         ppcno10.f\
         ppcno11.f\
         ppcno12Be.f\
         ppcno12BeBFe.f\
         ppcno12Li.f\
         ppcno12.f\
         ppcno3a12Ne.f\
         ppcno3a9.f\
         ppcno3ac10.f\
         ppcno9.f\
         ppcno9Fe.f\
         rq_reac.f\
         tabul_nuc.f\
         taux_nuc.f\
         vent.f\
         mod_kind.$(mod_ext)\
         mod_numerique.$(mod_ext)\
         mod_donnees.$(mod_ext)\
         mod_variables.$(mod_ext)\
         mod_nuc.f
$(FC90) $(FFLAGS90) -c $*.f

mod_bp_for_alecian.o mod_bp_for_alecian.$(mod_ext): \
```

```
         mod_donnees.$(mod_ext)\
         mod_variables.$(mod_ext)\
         mod_bp_for_alecian.f
$(FC90) $(FFLAGS90) -c $*.f

mod_evol.o mod_evol.$(mod_ext): \
         alecian1.f\
         coeff_rota.f\
         coeff_rota_saha.f\
         coeff_rota_z16.f\
         collision.f\
         coulomb.f\
         diffm.f\
         diffm_br.f\
         diffm_mp.f\
         difft.f\
         difft_gab.f\
         difft_nu.f\
         diffus.f\
         diffw.f\
         diffw_mpz.f\
 diffw_p03.f\
         eq_diff_chim.f\
         eq_diff_rota.f\
         evol.f\
         f_rad.f\
 lmix.f\
         pertw.f\
         pertw_loc.f\
         pertw_ptm.f\
         pertw_sch.f\
         resout_chim.f\
         resout_rota.f\
         rk_imps.f\
         mod_kind.$(mod_ext)\
         mod_numerique.$(mod_ext)\
         mod_donnees.$(mod_ext)\
         mod_variables.$(mod_ext)\
         mod_etat.$(mod_ext)\
         mod_nuc.$(mod_ext)\
         mod_bp_for_alecian.$(mod_ext)\
         mod_evol.f
$(FC90) $(FFLAGS90) -c $*.f

mod_static.o mod_static.$(mod_ext): \
         coll_qs.f\
         dgrad.f\
```

```
        lim_zc.f\
        pertm.f\
        pertm_ext.f\
        pertm_msol.f\
        pertm_tot.f\
        pertm_waldron.f\
        resout.f\
        static.f\
        static_m.f\
        static_r.f\
        thermo.f\
        update.f\
        mod_kind.$(mod_ext)\
        mod_numerique.$(mod_ext)\
        mod_donnees.$(mod_ext)\
        mod_variables.$(mod_ext)\
        mod_etat.$(mod_ext)\
        mod_opa.$(mod_ext)\
        mod_conv.$(mod_ext)\
        mod_atm.$(mod_ext)\
        mod_nuc.$(mod_ext)\
        mod_evol.$(mod_ext)\
        mod_static.f
$(FC90) $(FFLAGS90) -c $*.f

mod_cesam.o mod_cesam.$(mod_ext): \
        add_ascii.f\
        ascii.f\
        cesam.f\
        des.f\
        des_m.f\
        des_r.f\
        dnunl.f\
        initialise_rota.f\
        list.f\
        output.f\
        osc_adia.f\
        osc_invers.f\
        osc_nadia.f\
        u_initial.f\
        w_initial.f\
        mod_kind.$(mod_ext)\
        mod_numerique.$(mod_ext)\
        mod_donnees.$(mod_ext)\
        mod_variables.$(mod_ext)\
        mod_etat.$(mod_ext)\
        mod_opa.$(mod_ext)\
```

```
        mod_atm.$(mod_ext)\
        mod_nuc.$(mod_ext)\
        mod_static.$(mod_ext)\
        mod_cesam.f
$(FC90) $(FFLAGS90) -c $*.f

mod_exploit.o mod_exploit.$(mod_ext): \
        add_ascii.f\
        ascii.f\
        ctes_85.f\
        ctes_94.f\
        diffw.f\
        diffw_mpz.f\
 diffw_p03.f\
        ini_ctes.f\
        inter_atm.f\
        lit_binaire.f\
        lit_hr.f\
        lit_nl.f\
        min_max.f\
 min_max_cond.f\
        osc_adia.f\
        osc_invers.f\
        osc_nadia.f\
        output.f\
        read_ascii.f\
        write_nl.f\
        mod_kind.$(mod_ext)\
        mod_numerique.$(mod_ext)\
        mod_donnees.$(mod_ext)\
        mod_variables.$(mod_ext)\
        mod_etat.$(mod_ext)\
        mod_opa.$(mod_ext)\
        mod_nuc.$(mod_ext)\
        mod_exploit.f
$(FC90) $(FFLAGS90) -c $*.f


clean:
$(RM) cesam2k.out libcesam2k.a $(OBJS)  $(MODS)

touch:
touch $(SRCS)
```

Annexe B

Données diverses

On donne la liste des tableaux `glob` et `var` créés par `cesam`, *cf.* § 8.10 (Page 147), pour les sorties ASCII, ainsi que celle du tableau `list_cesam` servant à la compilation et à la création du module exécutable, *cf.* § 1.6 (Page 4).

B.1 Liste du tableau `glob`

```
glob(1)=mstar*msol
glob(2)=rtot*rsol
glob(3)=ltot*lsol
glob(4)=z0
glob(5)=x0
glob(6)=alpha
glob(7)=9./4.
glob(8)=1./162.
glob(9)=X dans ZC
glob(10)=Y dans ZC
glob(11)=d2p
glob(12)=d2ro
glob(13)=age
glob(14)=wrot vitesse de rotation globale
glob(15)=w_rot initial

nglob=15
```

B.2 Liste du tableau `var`

```
var(1,i): Rayon
var(2,i): Ln M/Mtot
var(3,i): Température
var(4,i): Pression
var(5,i): Densité
var(6,i): Gradient
var(7,i): Gradient
```

```
var(8,i): Luminosité
var(9,i): Opacité
var(10,i): Energie nuc+grav
var(11,i): Grand Gamma1
var(12,i): Gradient adiabatique
var(13,i): Delta
var(14,i): Cp
var(15,i): Mue^(-1)
var(16,i): Mu
var(17,i): Vaissala
var(18,i): Omega
var(19,i): dln kappa/dln T
var(20,i): dln kappa/dln ro
var(21,i): d epsilon(nuc) / d ln T
var(22,i): d epsilon(nuc) / d ln ro
var(23,i): !Ptot/Pgaz
var(24,i): !Gradient radiatif
var(25,i): d Gamma1 / d lnP (TY)
var(26,i): d Gamma1 / d lnT (PY)
var(27,i): d Gamma1 / dY (PT)
var(28,i): dP / dro (TX)
var(29,i): dP / dT (roX)
var(30,i): dP / dX (Tro)
var(31,i): du / dro (TX)
var(32,i): du / dT (roX)
var(33,i): du / dX(Tro)
var(34,i): énergie interne
var(35,i): d^2P / dro^2 (TX)
var(36,i): d^2P / dro dT (X)
var(37,i): d^2P / dT^2(roX)
var(38,i): d^2U / dro^2 (TX)
var(39,i): d^2U / dro dT (X)
var(40,i): d^2U / dT^2 (X)
var(41,i): dK / dX
var(42,i): d^2K / dT^2
var(43,i): d epsi / dX
var(44,i): dX / dR
var(45,i): J-B
var(46,i): Edding. facteur
var(ivar+j,i): xchim1g(j=1,nbelem) Abondances / gramme

ivar=46
```

B.3 Liste du tableau `list_cesam`

```
mod_kind
```

```
mod_numerique
mod_donnees
mod_variables
mod_etat
mod_opa
mod_conv
mod_atm
mod_nuc
mod_bp_for_alecian
mod_evol
mod_static
mod_cesam
mod_exploit
```

Annexe C

Liste des fichiers binaires

Les Fichiers binaires n'ayant pour fonction que des exploitations internes, leur contenu n'est pas détaillé. A l'exception du fichier binaire d'atmosphère `mon_modele.atm`, ces fichiers peuvent être lus par la routine `lit_binaire`, *cf.* § 8.71 (Page 177).

- Fichier binaire du modèle de pré-séquence principale homogène : `mon_modele_B.pms`.
- Fichier binaire du modèle de séquence principale d'âge zéro homogène : `mon_modele_B.hom`.
- Fichier binaire du modèle de séquence principale d'âge zéro : `mon_modele_B.zams`.
- Fichier binaire du modèle de post-séquence principale : `mon_modele_B.post` ou `mon_modele_B.tams`.
- Fichier binaire du modèle du début de la combustion de l'hélium : `mon_modele_B.cohe`.
- Fichier binaire du modèle du début de la combustion du carbone : `mon_modele_B.coca`.
- Fichier binaire du modèle final : `mon_modele_B.dat`.
- Fichier binaire du modèle intermédiaire : `mon_modele_B.rep`.
- Fichier binaire du modèle d'atmosphère : `mon_modele_B.atm`.

Annexe D

Liste des fichiers ASCII

Se reporter § 3.3 (Page 22) pour la description du fichier ASCII de données `mon_modele.don`.

D.1 Fichiers pour oscillations

Les fichiers ASCII pour les oscillations ont une en-tête commune. Leur contenu diffère ensuite suivant leur type.

`en-tête` :

- Lignes 1-4 : identification et physique utilisée.
- Ligne 5 : nombre, noms des éléments chimiques utilisés.
- Ligne 5 : nombre de couches, de "global" (13), de variables, d'éléments chimiques (redondant), indice de la vitesse angulaire (s'il n'en est pas tenu compte, l'indice est -1).

Exemple :

```
Fichier pour inversion: test-inv.osc
CESAM2k version 0.0.0.0 lagr colloc  1  2 np no diffus, 31 Aout 2003 17h45
Physique utilisée: etat_eff, opa_int_zsx, conv_jmj, ppcno9, NACRE
solaire_gn, lim_atm, hopf, perte_ext, diffm_mp, difft_nu, ctes_94
 10  H1  He3  He4  C12  C13  N14  N15  O16  O17  Si28
      472        13        25        10        -1
 1.989190000000E+33 6.959888774683E+10 3.846010241481E+33 1.723296157963E-02 7.033715921640E-01
 1.601359931639E+00 7.033715881909E-01 2.793954502293E-01-1.234804262016E+02-1.250212463069E+02
 4.650000000000E+03 0.000000000000E+00 0.000000000000E+00
 6.965125316795E+10 1.125188830934E-10 4.712514552821E+03 1.082443642936E+03 3.550000000000E-09
 2.317104979700E-04 3.846010241481E+33 5.992732100399E-03 0.000000000000E+00 1.664711685203E+00
 3.986098681160E-01 1.004603449555E+00 1.630687926937E+08 1.300571860485E+00 2.500897748491E+03
 0.000000000000E+00 1.729836584721E+00 8.429278614488E-01 0.000000000000E+00 0.000000000000E+00

........................................

 3.983397251906E-01 9.536638658353E-01 2.457927862174E+08 8.289918269170E-01 0.000000000000E+00
 0.000000000000E+00-1.948967682005E+00 3.878822249046E-01 2.023289474023E+02 1.828866051074E+01
 1.000000000000E+00 3.288637435509E-01-5.586998681649E-03 6.642708700530E-03-8.894901969547E-03
 3.750487629847E-01 1.039826643345E-05 6.071493899575E-01 1.637161572818E-05 4.475726236722E-06
 4.374234929779E-03 1.935496963607E-07 7.968745830613E-03 3.970500883875E-04 5.030377050940E-03
```

D.1.1 Fichier pour oscillations adiabatiques

```
glob: variables globales du fichier mon_modele-ad.osc

glob(1)=mstar*msol
glob(2)=rtot*rsol
glob(3)=ltot*lsol
glob(4)=z0
glob(5)=x0
glob(6)=alpha
glob(7)=X dans ZC
glob(8)=Y dans ZC
glob(9)=d2p
glob(10)=d2ro
glob(11)=age
glob(12)=w_rot initial

var: variables locales utilisées ; nvar=22 pour oscillations adiabatiques

var(1,i)=r*rsol
var(2,i)=log(m/mstar) -1.d38 au centre
var(3,i)=t
var(4,i)=Ptot
var(5,i)=ro
var(6,i)=gradient réel d ln T / d ln P
var(7,i)=l
var(8,i)=kap
var(9,i)=énergie thermo+gravifique
var(10,i)=grand Gamma1
var(11,i)=gradient adiabatique
var(12,i)=delta
var(13,i)=cp
var(14,i)=1 / (mu elec.)
var(15,i)=vaissala, 0 au centre
var(16,i)=vitesse angulaire, radian/sec
var(17,i)=d ln kappa / d ln T
var(18,i)=d ln kappa / d ln ro
var(19,i)=d epsilon(nuc) / d ln T
var(20,i)=d epsilon(nuc) / d ln ro
var(21,i)=Ptot / Pgas
var(22,i)=gradient radiatif

composition chimique

var(22+j,i)=xchim(j)*nucleo(j), j=1,nbelem
```

D.1.2 Fichier pour oscillations non adiabatiques

```
glob: variables globales du fichier mon_modele-nad.osc

var: variables
glob(1)=mstar*msol
glob(2)=rtot*rsol
glob(3)=ltot*lsol
glob(4)=z0
glob(5)=x0
glob(6)=alpha
glob(7)=X dans ZC
glob(8)=Y dans ZC
glob(9)=d2p
glob(10)=d2ro
glob(11)=age
glob(12)=wrot vitesse de rotation globale
glob(13)=w_rot initial

var: variables locales utilisées ; nvar=44 pour oscillations non adiabatiques

var(1,i)=r*rsol
var(2,i)=log(m/mstar) -1.d38 au centre
var(3,i)=t
var(4,i)=Ptot
var(5,i)=ro
var(6,i)=gradient reel d ln T / d ln P
var(7,i)=l
var(8,i)=kap
var(9,i)=énergie thermo+gravifique
var(10,i)=grand Gamma1
var(11,i)=gradient adiabatique
var(12,i)=delta
var(13,i)=cp
var(14,i)=1 / (mu elec.)
var(15,i)=vaissala, 0 au centre
var(16,i)=vitesse angulaire, radian/sec
var(17,i)=d ln kappa / d ln T
var(18,i)=d ln kappa / d ln ro
var(19,i)=d epsilon(nuc) / d ln T
var(20,i)=d epsilon(nuc) / d ln ro
var(21,i)=Ptot / Pgas
var(22,i)=gradient radiatif
var(23,i)=d Gamma1 / d log P
var(24,i)=d Gamma1 / d log T
var(25,i)=d Gamma1 / dY = d Gamma1 / dZ
var(26,i)=dP / dro (TX)
```

```
var(27,i)=dP / dT (roX)
var(28,i)=dP / dX (Tro)
var(29,i)=du / dro (TX)
var(30,i)=du / dT (roX)
var(31,i)=du / dx(Tro)
var(32,i)=énergie interne
var(33,i)=d^2P / dro^2 (TX)
var(34,i)=d^2P / dro dT (X)
var(35,i)=d^2P / dT^2(roX)
var(36,i)=d^2U / dro^2 (TX)
var(37,i)=d^2U / dro dT (X)
var(38,i)=d^2U / dT^2 (X)
var(39,i)=dK / dX
var(40,i)=d^2K / dT^2
var(41,i)=d epsi / dX
var(42,i)=dX / dR
var(43,i)=J-B
var(44,i)=Edding. facteur

composition chimique

var(44+j,i)=xchim(j)*nucleo(j), j=1,nbelem
```

D.1.3 Fichier pour inversions

```
glob: variables globales du fichier mon_modele-inv.osc

glob(1)=mstar*msol
glob(2)=rtot*rsol
glob(3)=ltot*lsol
glob(4)=z0
glob(5)=x0
glob(6)=alpha
glob(7)=X dans ZC
glob(8)=Y dans ZC
glob(9)=d2p
glob(10)=d2ro
glob(11)=age
glob(12)=wrot vitesse de rotation globale
glob(13)=w_rot initial

var: variables locales utilisées ; nvar=25 pour inversion

var(1,i)=r*rsol
var(2,i)=log(m/mstar) -1.d38 au centre
var(3,i)=t
```

```
var(4,i)=Ptot
var(5,i)=ro
var(6,i)=gradient reel d ln T / d ln P
var(7,i)=l
var(8,i)=kap
var(9,i)=énergie thermo+gravifique
var(10,i)=grand Gamma1
var(11,i)=gradient adiabatique
var(12,i)=delta
var(13,i)=cp
var(14,i)=1 / (mu elec.)
var(15,i)=vaissala, 0 au centre
var(16,i)=vitesse angulaire, radian/sec
var(17,i)=d ln kappa / d ln T
var(18,i)=d ln kappa / d ln ro
var(19,i)=d epsilon(nuc) / d ln T
var(20,i)=d epsilon(nuc) / d ln ro
var(21,i)=Ptot / Pgas
var(22,i)=gradient radiatif
var(23,i)=d Gamma1 / d log P
var(24,i)=d Gamma1 / d log T
var(25,i)=d Gamma1 / dY = d Gamma1 / dZ

composition chimique

var(25+j,i)=xchim(j)*nucleo(j), j=1,nbelem
```

D.1.4 Utilisation du nombre maximum de couches

Avec certains types de précision, *cf.* § 3.3 (Page 22), il est possible de créer un fichier ASCII d'oscillation comportant au moins un nombre de couches fixé à l'avance ; le nombre maximum de couches étant utilisé lors du calcul des derniers modèles. Les critères requis pour ce faire sont les suivants :

- Type de précision utilisé : 'sa', ou 'co'.
- ***ET*** réalisation d'un des critères suivants :
  - Age du modèle à calculer inférieur d'un million d'années de l'âge maximum à atteindre.
  - Numéro du modèle à calculer égal à `NMAX_MODELS` – 1.
  - Température effective voisine de la limite demandée.
  - Abondance centrale d'hydrogène voisine de la limite demandée.
  - Extension du noyau d'hélium voisine de la limite demandée.

D.2 Fichier pour diagramme HR

Le fichier pour tracé/exploitation du diagramme HR est systématiquement implémenté au cours de chaque exécution de CESAM2k. Il est créé lors de l'initialisation d'un modèle sur la séquence principale d'âge zéro homogène ou la pré-séquence principale homogène. Il est complété en cas de reprise d'un modèle existant.

Pour chaque modèle, sur une première ligne sont reportés :

1. l'âge en 10^6 ans ;
2. le nombre d'éléments du vecteur de composition chimique ;
3. le nombre de limites zones radiatives / zones convectives
4. le numéro du modèle,
5. pour chaque limite zone radiative / zone convective la mention "T" ou "F" suivant qu'il s'agit, ou non, du début d'une zone convective ;
6. suivant le nombre de limites, une ou plusieurs lignes sont nécessaires pour indiquer :
   (a) $\log T_{\text{eff}}$,
   (b) $\log L/L_\odot$,
   (c) $\log R/R_\odot$
   (d) pour chaque limite, $M_\star/M_\odot$, $(M_\star - m)/M_\odot$, $r/R_\odot$, $r_{\text{ov}}/R_\odot$, (r_{ov} étant le rayon de la limite étendue par overshooting).

Dans les cas particuliers, on utilise les conventions suivantes :

– modèle totalement convectif : 1 (seule) limite "F" placée au centre ($r/R_\odot = 0$),

– modèle totalement radiatif : 0 (aucune) limite "F" placée en -100 ($r/R_\odot = -100$).

Viennent ensuite les noms des éléments chimiques et leurs abondances, par unité de masse, au centre et à la surface. S'il n'est pas imposé d'overshoot, les valeurs r_{ov} des rayons des limites étendues sont fixées à -100. S'il est tenu compte de la diffusion du moment cinétique, une ligne supplémentaire, identifiée pr `Wrot` indique les vitesses linéaire et angulaire de la couche externe. Exemple :

```
 0.000000000000000E+00 10  2   0 F T
 3.749146E+00-1.723498E-01-6.098130E-02 1.000000E+00 9.053737E-01 1.262897E-01
-1.000000E+02 2.969686E-02 6.310311E-01-1.000000E+02
 H1  6.97718E-01 6.97718E-01
He3  8.91487E-05 8.91487E-05
He4  2.82700E-01 2.82700E-01
C12  3.33842E-03 3.33842E-03
C13  4.02357E-05 4.02357E-05
N14  1.03227E-03 1.03227E-03
N15  4.06226E-06 4.06226E-06
O16  9.39680E-03 9.39680E-03
O17  3.80402E-06 3.80402E-06
Si28 5.67649E-03 5.67649E-03
Wrot 5.00050E+01 4.92437E-05  <-- qu'avec diffusion du moment cinétique
 1.000000000000000E+01 10  2   1 F T
 3.750235E+00-1.660591E-01-5.995672E-02 1.000000E+00 9.485921E-01 1.002733E-01
```

```
-1.000000E+02 2.900062E-02 6.330032E-01-1.000000E+02
 H1  6.97097E-01 6.97795E-01
He3  6.62006E-05 8.91324E-05
He4  2.83225E-01 2.82627E-01
C12  2.45937E-03 3.33777E-03
C13  4.11117E-04 4.02273E-05
N14  1.66274E-03 1.03208E-03
N15  2.18466E-07 4.06144E-06
O16  9.39741E-03 9.39506E-03
O17  3.87613E-06 3.80329E-06
Si28 5.67687E-03 5.67553E-03
Wrot 5.21773E+01 5.03611E-05  <-- qu'avec diffusion du moment cinétique
..................................

 4.685000000000000E+03 10  31  1 T
 3.761741E+00-3.270666E-06 2.093004E-06 1.000000E+00 2.259322E-02 7.173108E-01
-1.000000E+02
 H1  3.22007E-01 7.30094E-01
He3  6.46882E-06 8.19412E-05
He4  6.57151E-01 2.51936E-01
C12  2.05912E-05 3.04480E-03
C13  5.63346E-06 3.64768E-05
N14  5.13761E-03 9.44078E-04
N15  2.25834E-07 3.69734E-06
O16  8.99719E-03 8.61328E-03
O17  7.99800E-04 3.47279E-06
Si28 5.87454E-03 5.24189E-03
Wrot 1.94190E+02 8.24763E-05  <-- qu'avec diffusion du moment cinétique
```

Les fichiers pour diagramme HR sont lus par la routine `lit_hr` du module `mod_exploit`.

Annexe E

Liste détaillée des modules

Afin de faciliter la lecture des algorithmes de CESAM2k, on donne la liste détaillée des variables et routines des modules, en indiquant l'endroit de leur initialisation ; celles-ci sont souvent effectuées dans des routines dépendantes, *cf.* § 8.1 (Page 141), par raison de concision, seuls les noms des routines génériques sont mentionnés. Les routines étant par ailleurs largement décrites, elles ne sont ci-après que citées. Chaque routine `PRIVATE` et/ou `PUBLIC` est introduite sous la forme d'`INCLUDE` dans son module d'appartenance. Ci-après, la description des fonctions des variables logiques correspond à leur valeur .TRUE. L'ordre de présentation est celui de la compilation indiqué dans le fichier `list_cesam`,*cf.* § B.3 (Page 234), du sous-dirctory `SOURCE`.

E.1 Module mod_kind

Le module `mod_kind` regroupe les types des variables.

– Paramètres PUBLIC :

dp : Définit les variables en double précision.

sp : Définit les variables en simple précision.

E.2 Module mod_numerique

Le module `mod_numerique` regroupe les routines purement numériques.

– Variable PUBLIC logical :

no_croiss : La suite des abscisses n'est pas strictement croissante, ou encore une autre difficulté est survenue lors de l'éxécution.

– Routines PRIVATE :

bval0, colpnt, difdiv, horner, schu58_n.

– Routines PUBLIC :

arb_rom, boite, box, bsp1ddn, bsp1dn, bsp_dis, bvald, bval1, coll, delete_doubles, fermi_dirac, gauss_band, genere_bases, intgauss, linf, matinv, min_max, neville, newspl, newton, noedif, noein, noeud, noeu_dis, pause, polyder, shell, sum_n, zoning

E.3 Module mod_donnees

Le module `mod_donnees` regroupe l'ensemble des quantités fixes au cours de l'évolution.

- Paramètres PUBLIC double précision :

  d_conv : Coefficient de diffusion dans les zones mélangées.

  dtmin : Pas temporel minimum en Myrs.

  dx_tams : Précision sur l'abondance en X à la fin de la séquence principale d'âge zéro.

  phi : $d\ln\rho/d\ln\mu$.

  x_tams : Abondance en X définissant la fin de la séquence principale d'âge zéro.

- Paramètres PUBLIC integer :

  n_min : Nombre minimum de couches.

  pnzc : Nombre maximum de zones convectives pouvant être prises en compte.

  r_qs : Ordre des équations différentielles des équations d'équilibre quasi-statique.

- Paramètres PUBLIC character :

  version : Numéro de version lu par un INCLUDE du fichier `journal`, *cf.* § 1.2 (Page 2).

- Variables PUBLIC double précision :

  ab_ini : Abondances initiales, initialisé dans `nuc.f`.

  ab_min : Abondances minimales, initialisé dans `nuc.f`.

  nucleo : Masses atomiques des isotopes, initialisé dans `tabul_nuc.f`.

  rot_min : Valeurs négligeables des variables de la diffusion du moment cinétique, initialisé dans `lit_nl.f`.

  xvent : Composition chimique du vent, initialisé dans `vent.f`.

  zi : Charges des isotopes, initialisé dans `tabul_nuc.f`.

  abon_m : Abondances initiales par masse des éléments chimiques, initialisé dans `abon_ini.f`.

  abe7 : Masse atomique du ^{7}Be, initialisé dans `taux_nuc.f`.

  abe9 : Masse atomique du ^{9}Be, initialisé dans `taux_nuc.f`.

  ab11 : Masse atomique du ^{11}B, initialisé dans `taux_nuc.f`.

  ac12 : Masse atomique du ^{12}C, initialisé dans `taux_nuc.f`.

  ac13 : Masse atomique du ^{13}C, initialisé dans `taux_nuc.f`.

  afe56 : Masse atomique du ^{56}Fe, initialisé dans `taux_nuc.f`.

  af18 : Masse atomique du ^{18}F, initialisé dans `taux_nuc.f`.

  af19 : Masse atomique du ^{19}F, initialisé dans `taux_nuc.f`.

  agemax : Age maximum à atteindre, lu dans `lit_nl.f`.

ah : Masse atomique du ^{1}H, initialisé dans `taux_nuc.f`.

ah2 : Masse atomique du ^{2}H, initialisé dans `taux_nuc.f`.

ahe3 : Masse atomique du ^{3}He, initialisé dans `taux_nuc.f`.

ahe4 : Masse atomique du ^{4}He, initialisé dans `taux_nuc.f`.

ali6 : Masse atomique du ^{6}Li, initialisé dans `taux_nuc.f`.

ali7 : Masse atomique du ^{7}Li, initialisé dans `taux_nuc.f`.

alpha : Longueur de mélange, lu dans `lit_nl.f`.

amg23 : Masse atomique du ^{23}Mg, initialisé dans `taux_nuc.f`.

amg24 : Masse atomique du ^{24}Mg, initialisé dans `taux_nuc.f`.

amg25 : Masse atomique du ^{25}Mg, initialisé dans `taux_nuc.f`.

amg26 : Masse atomique du ^{26}Mg, initialisé dans `taux_nuc.f`.

amu : Masse atomique unité m_u, initialisé dans `taux_nuc.f`.

an : Masse atomique du neutron, initialisé dans `taux_nuc.f`.

ana23 : Masse atomique du ^{23}Na, initialisé dans `taux_nuc.f`.

ane20 : Masse atomique du ^{20}Ne, initialisé dans `taux_nuc.f`.

ane21 : Masse atomique du ^{21}Ne, initialisé dans `taux_nuc.f`.

ane22 : Masse atomique du ^{22}Ne, initialisé dans `taux_nuc.f`.

an13 : Masse atomique du ^{13}N, initialisé dans `taux_nuc.f`.

an14 : Masse atomique du ^{14}N, initialisé dans `taux_nuc.f`.

an15 : Masse atomique du ^{15}N, initialisé dans `taux_nuc.f`.

ap : Masse atomique du proton, initialisé dans `taux_nuc.f`.

ao16 : Masse atomique du ^{16}O, initialisé dans `taux_nuc.f`.

ao17 : Masse atomique du ^{17}O, initialisé dans `taux_nuc.f`.

ao18 : Masse atomique du ^{18}O, initialisé dans `taux_nuc.f`.

aradia : Constante de la radiation a, initialisé dans `ini_ctes.f`.

clight : Célérité de la lumière c, initialisé dans `ini_ctes.f`.

cpturb : Paramètre de pression turbulente, lu dans `lit_nl.f`.

ctel : Facteur de répartition en luminosité, initialisé dans `cesam.f`.

ctem : Facteur de répartition en masse, initialisé dans `cesam.f`.

ctep : Facteur de répartition en pression, initialisé dans `cesam.f`.

cter : Facteur de répartition en rayon, initialisé dans `cesam.f`.

ctet : Facteur de répartition en température, initialisé dans `cesam.f`.

dpsi : Taux de variation maximale pour la constante de répartition entaînant une modification du nombre de couches `n_qs` dans le cas standard d'une grille ajustable, initialisé dans `cesam.f`.

dn_fixe : Taux de variation maximale pour la constante de répartition entaînant une modification du nombre de couches `nc_fixe` dans le cas d'une grille fixe, initialisé dans `cesam.f`.

dtlist : Intervalle de temps entre deux listings complets, lu dans `lit_nl.f`.

dtmax : Pas temporel maximal, initialisé dans `cesam.f`.

dt0 : Pas temporel initial, initialisé dans `cesam.f`.

d_grav : Variation maximale de l'énergie graviphique sur un pas temporel, initialisé dans `cesam.f`.

d_turb : Coefficient de diffusion turbulente, lu dans `lit_nl.f`.

echarg : Charge de l'électron, initialisé dans `ini_ctes.f`.

eve : Energie de l'électronvolt, initialisé dans `ini_ctes.f`.

fesh_sol : Valeur du [Fe/H] solaire, initialisé dans `ini_ctes.f` .

g : Constante de la gravitation G, initialisé dans `ini_ctes.f`.

gmsol : Produit $G \times M_\odot$, initialisé dans `ini_ctes.f`.

granr : Constante des gaz parfaits $\mathcal{R}$, initialisé dans `ini_ctes.f`.

he_core : Masse du noyau d'hélium à partir de laquelle l'évolution sera arrêtée, lu dans `lit_nl.f`.

hhe_core : Masse réduite du noyau d'hélium à partir de laquelle l'évolution sera arrêtée , initialisé dans `lit_nl.f`.

hpl : Constante de Plank h, initialisé dans `ini_ctes.f`.

kbol : Constante de Boltzman k, initialisé dans `ini_ctes.f`.

lbol0 : Zéro des magnitudes bolométriques, initialisé dans `ini_ctes.f`.

li_ini : Abondance de ^{7}Li dans le noyau d'un modèle de séquence principale d'âge zéro homogène, initialisé dans `abon_ini.f`.

lnt_stop : L'évolution est arrêtée lorsque cette température effective, en ln, est franchie par valeurs croissantes ou décroissantes, initialisé dans `lit_nl.f` .

ln_Tli : Température au dessus de laquelle il n'y a pas de lithium inital pour le modèle de séquence principale d'âge zéro, initialisé dans `abon_ini.f`.

ln10 : ln(10), initialisé dans `ini_ctes.f`.

loc_zc : Paramètre de localisarion des limites zones radiatives / zones convectives, initialisé dans `cesam.f`.

log_teff : L'évolution est arrêtée lorsque cette température effective est franchie par valeurs croissantes ou décroissantes, lu dans `lit_nl.f`.

lsol : Luminosité solaire $L_\odot$, initialisé dans `ini_ctes.f`.

mdot : Taux de perte de masse, lu dans `lit_nl.f`.

me : Masse de l'électron m_e, initialisé dans `ini_ctes.f`.

mterre : Masse de la terre, initialisé dans `ini_ctes.f`.

msol : Masse solaire $M_\odot$, initialisé dans `ini_ctes.f`.

mtot : Masse initiale du modèle, lu dans `lit_nl.f`.

ovshti : Taux d'overshoot inférieur, lu dans `lit_nl.f`.

ovshts : Taux d'overshoot supérieur, lu dans `lit_nl.f`.

pi : π, initialisé dans `ini_ctes.f`.

precit : Paramètre de précision de l'intégration temporelle de la composition chimique, initialisé dans `cesam.f`.

precix : Paramètre de précision de l'intégration spatiale de l'équilibre quasi-statique, initialisé dans `cesam.f`.

p_pertw : Paramètre de perte de moment cinétique, lu dans `lit_nl.f`.

re_nu : Taux de turbulence radiative, lu dans `lit_nl.f`.

ro_test : Valeur minimale de la densité en dessous de laquelle la variation d'énergie graviphique n'est plus considérée comme significative, initialisé dans `cesam.f`.

rsol : Rayon solaire $R_\odot$, initialisé dans `ini_ctes.f`.

secon6 : Nombre de secondes par million d'années, initialisé dans `ini_ctes.f`.

sigma : Constante de Stéfan σ, initialisé dans `ini_ctes.f`.

tau_max : Profondeur optique de raccord atmosphère/enveloppe, lu dans `lit_nl.f`.

t_inf : Température minimale de tabulation des réactions thermonucléaires, initialisé dans `tabul_nuc.f`.

t_sup : Température maximale de tabulation des réactions thermonucléaires, initialisé dans `tabul_nuc.f`.

t_stop : Température centrale arrêtant le calcul de l'évolution, lu dans `lit_nl.f`.

w_rot : Vitesse angulaire initiale, lu dans `lit_nl.f`.

x0 : Abondance initiale par masse d'hydrogène X, lu ou déterminé dans `lit_nl.f`.

x_stop : Abondance centrale d'hydrogène arrêtant le calcul de l'évolution, lu dans `lit_nl.f`.

y0 : Abondance initiale par masse d'hélium Y, lu dans `lit_nl.f`.

zsx_sol : Z/X solaire, initialisé dans `ini_ctes.f`.

zsx0 : Rapport initial Z/X, lu ou initialisé dans `lit_nl.f` .

z0 : Abondance initiale par masse des métaux Z, initialisé dans `lit_nl.f`.

– Variables PUBLIC simple précision :

dfesh_des : Barres d'erreur pour le dessin de [Fe/H], initialisé dans `des.f`.

dl_des : Barres d'erreur pour le dessin de $\log(L)$, initialisé dans `des.f`.

dteff_des : Barres d'erreur pour le dessin de $\log T_{\rm eff}$, initialisé dans `des.f`.

zoom_l : Extensions asymétriques en luminosité, initialisé dans `des.f`.

zoom_t : Extensions asymétriques en Teff, initialisé dans `des.f`.

dh : Espace en hauteur entre cadres des dessins, initialisé dans `des.f`.

dl : Espace en largeur entre cadres des dessins, initialisé dans `des.f`.

h : Hauteur des cadres des dessins, initialisé dans `des.f`.

ld : Largeur des cadres des dessins, initialisé dans `des.f`.

fesh_des : Cible en [Fe/H] en surface, initialisé dans `des.f`.

l_des : Cible en $L/L_\odot$, initialisé dans `des.f`.

teff_des : Cible en température, initialisé dans `des.f`.

logteff_max : Limite à gauche des abscisses du cadre du diagramme HR, initialisé dans `des.f`.

logteff_min : Limite à droite des abscisses du cadre du diagramme HR, initialisé dans `des.f`.

logl_max : Limite supérieure des ordonnées du cadre du diagramme HR, initialisé dans `des.f`.

logl_min : Limite inférieure des ordonnées du cadre du diagramme HR, initialisé dans `des.f`.

xleft : Limite à droite des dessins, initialisé dans `des.f`.

ybot : Limite inférieure des dessins, initialisé dans `des.f`.

y_age : Ordonnées pour l'écriture de l'âge au dessus des cadres des dessins, initialisé dans `des.f`.

– Variables PUBLIC integer :

Krot : Flag d'identification du type de rotation, initialisé dans `lit_nl`.

ife56 : Indice de l'isotope ^{56}Fe, initialisé dans `nuc.f`.

ihe4 : Indice de l'isotope ^{4}He, initialisé dans `nuc.f`.

ini0 : Nombre maximum d'itérations globales avec réestimation de la composition chimique et de la délimitation des limites zones radiatives / zones convectives, initialisé dans `cesam.f`.

Ipg : Indice de la variable pression gazeuse, initialisé dans `cesam.f`.

i_ex : Indice de l'élément fictif, initialisé dans `nuc.f`.

m_ch : Ordre des B-splines d'interpolation de la composition chimique, initialisé dans `cesam.f`.

m_ptm : Ordre des B-splines d'interpolation de la perte de masse, initialisé dans `cesam.f`.

m_qs : Ordre des B-splines pour la résolution de l'équilibre quasi-statique, initialisé dans `cesam.f`.

m_rot : Ordre des B-splines d'interpolation du moment cinétique, initialisé dans `cesam.f`.

m_tds : Ordre des B-splines d'interpolation de l'énergie graviphique, initialisé dans `cesam.f`.

nb_max_modeles : Nombre maximum de modèles à calculer, lu dans `lit_nl.f`.

nchim : Nombre d'isotopes, initialisé dans `nuc.f`.

ne : Nombre de variables pour l'équilibre quasi-statique,initialisé dans `cesam.f`.

nrot : Nombre de variables pour la diffusion du moment cinétique, initialisé dans `lit_nl.f`.

n_atm : Nombre de couches dans l'atmosphère, initialisé dans `cesam.f`.

n_max : Nombre maximum de couches pour l'équilibre quasi-statique, initialisé dans `cesam.f`.

ordre : Ordre d'intégration des équations d'évolution de la composition chimique sans diffusion, initialisé dans `cesam.f`.

ord_qs : Ordre des B-splines pour l'intégration des équations d'équilibre quasi-statique, initialisé dans `cesam.f`.

– Variables PUBLIC logical :

all_output : On garde dans l'environnement tous les fichiers binaires de reprise dont le nom comporte le numéro du modèle ; initialisé dans `lit_nl.f`.

all_rep : On garde dans l'environnement tous les fichiers ASCII dont le nom comporte le numéro du modèle ; initialisé dans `lit_nl.f`.

diffusion : On tient compte de la diffusion microscopique des isotopes, lu dans `lit_nl.f`.

ecrit_rot : Ecriture du ficheir permettant le tracé des variables et des coefficients de la diffusion du moment cinétique.

en_masse : Utilisation des variables lagrangiennes, lu dans `lit_nl.f`.

garde_xish : On conserve les rapports métal/hydrogène de la mixture initiale, lu dans `lit_nl.f`.

grad_ovi : On utilise le gradient adiabatiaque dans les extensions inférieures des zone convective ; initialisé dans `lit_nl.f`.

grad_ovs : On utilise le gradient adiabatiaque dans les extensions supérieures des zone convective ; initialisé dans `lit_nl.f`.

grille_fixe : La grille d'interpolation de la composition chimique est fixe, lu dans `lit_nl.f`.

He_ajuste : On ajustera la limite du noyau d'hélium avant de sortir, initialisé dans `lit_nl.f`.

jpz : Utilisation du formalisme de pénétration convective de JP. Zahn, lu dans `lit_nl.f`.

kipp : Utilisation de l'approximation de Kippenhahan, lu dans `lit_nl.f`.

ledoux : Utilisation du critère de convection de Ledoux, lu dans `lit_nl.f`.

lim_ro : Utilisation de la limite en densité pour la restitution de l'atmosphère, lu dans `lit_nl.f`.

mitler : Utilisation du formalisme de Mitler pour le calcul de l'effet d'écran, lu dans `lit_nl.f`.

modif_chim : On personnalisera la composition chimique initiale, lu dans `lit_nl.f`.

mvt_dis : On tiendra compte des déplacements des discontinuités pour la détermination de la composition chimique, initialisé dans `cesam.f`.

pturb : On tiendra compte de la pression turbulente, initialisé dans `lit_nl`.

rep_atm : On utilisera un fichier binaire d'atmosphère, s'il existe, initialisé dans `cesam.f`.

rot_solid : On supposera la rotation solide, lu dans `lit_nl.f`.

t_ajuste : On ajustera la température centrale avant de sortir, initialisé dans `lit_nl.f`.

x_ajuste : On ajustera l'abondance centrale d'hydrogène avant de sortir, initialisé dans `lit_nl.f`.

– Variables PUBLIC character :

precision : Désignation du type de calcul, lu dans `lit_nl.f`.

nom_elem : Noms des isotopes utilisés, initialisé dans `tabul_nuc.f`.

nom_rot : Noms des variables utilisées pour la diffusion du moment cinétique, initialisé dans `lit_nl.f`.

arret : Désignation du type d'arrêt, lu dans `lit_nl.f`.

nom_xheavy : Nom de l'élément fictif, initialisé dans `nuc.f`.

unit : Nom de l'unité utilisée pour la donnée de la vitesse angulaire initiale, initialisé dans `lit_nl.f`.

langue : Nom de la langue utilisée pour les commentaires, initialisé dans `cesam.f` .

nom_atm : Nom du type d'atmosphère, lu dans `lit_nl.f`.

nom_abon : Nom des abondances initiales, lu dans `lit_nl.f`.

nom_conv : Nom du type de convection, lu dans `lit_nl.f`.

nom_ctes : Nom du groupe des constantes physiques, lu dans `lit_nl.f`.

nom_des : Nom du type de dessin, lu dans `lit_nl.f`.

nom_diffm : Nom du type de coefficients de diffusion microscopique, lu dans `lit_nl.f`.

nom_diffw : Nom du type de coefficients de diffusion de moment cinétique, lu dans `lit_nl.f`.

nom_difft : Nom du type de coefficients de diffusion turbulente à utiliser, lu dans `lit_nl.f`.

nom_etat : Nom du type d'équation d'état, lu dans `lit_nl.f`.

nom_frad : Nom du type d'accélérations radiatives, lu dans `lit_nl.f`.

nom_nuc : Nom du type de réseau de réactions thermonucléaires, lu dans `lit_nl.f`.

nom_nuc_cpl : Nom du type de compilation de réactions thermonucléaires, lu dans `lit_nl.f`.

nom_output : Nom du type de sortie ASCII, lu dans `lit_nl.f`.

nom_pertm : Nom du type de perte de masse, lu dans `lit_nl.f`.

nom_pertw : Nom du type de perte de moment cinétique, lu dans `lit_nl.f`.

nom_tdetau : Nom du type de loi $T(\tau)$, lu dans `lit_nl.f`.

nom_fich2 : Nom du fichier de sortie `mon_modele.lis`, lu dans `lit_nl.f`.

f_eos : Nom du fichier d'équation d'état, lu dans `lit_nl.f`.

f_opa : Nom du fichier d'opacité, lu dans `lit_nl.f`.

nom_opa : Nom du type d'opacité, lu dans `lit_nl.f`.

source : Nom de la source des constantes physiques, initialisé dans `ini_ctes.f`.

methode : Description de la méthode de calcul, initialisé dans `cesam.f`.

device : Nom du device de dessin, initialisé dans `des.f`.

nom_chemin : Nom du chemin du directory des données physiques, lu dans `lit_nl.f`.

– Routines PUBLIC :

lit_nl, ini_ctes, print_ctes.

– Routines PRIVATE :

ctes_85, ctes_94.

E.4 Module mod_variables

Ce module regroupe les routines d'usage général et les quantités variant au cours d'une évolution. Par exemple, le vecteur des variables quasi-statiques, ou encore le vecteur de composition chimique.

– Variables PUBLIC double précision :

bp : Table des coefficients des B-splines des variables quasi-statiques, calculée dans `coll_qs.f`.

bp_t : Table des coefficients des B-splines des variables quasi-statiques au pas temporel précédent, réinitialisé dans `update.f`.

chim : Table des coefficients des B-splines de la composition chimique, calculé dans `evol.f` ou `resout_diff_chim.f`.

chim_t : Table des coefficients des B-splines de la composition chimique au pas temporel précédent, réinitialisé dans `update.f`.

old_ptm : Tables des coefficients des B-splines des masses, calculé dans `pertm.f`.

rota : Table des coefficients des B-splines de la rotation, calculé dans `resout_rota.f`.

rota_t : Table des coefficients des B-splines de la rotation au pas temporel précédent, réinitialisé dans `update.f`.

tds : Table des coefficients des B-splines de l'énergie graviphique, calculé dans `cesam.f`.

tds_t : Table des coefficients des B-splines de l'énergie graviphique au pas temporel précédent, réinitialisé dans `update.f`.

mc : Table des masses pour l'interpolation de la composition chimique, calculé dans `evol.f` ou `diffus.f`.

mct : Vecteur nodal des masses pour l'interpolation de la composition chimique, calculé dans `evol.f` ou `diffus.f`.

mc_fixe : Table des masses de la grille fixe, calculé dans `evol.f`.

mc_t : Table des masses pour l'interpolation de la composition chimique au pas temporel précédent, réinitialisé dans `update.f`.

mct_t : Vecteur nodal des masses pour l'interpolation de la composition chimique au pas temporel précédent, réinitialisé dans `update.f`.

mrot : Table des masses pour l'interpolation du moment cinétique, calculé dans `diffus.f`.

mrott : Vecteur nodal des masses pour l'interpolation du moment cinétique, calculé dans `diffus.f`.

mrott_t : Vecteur nodal des masses pour l'interpolation du moment cinétique au pas temporel précédent, réinitialisé dans `update.f`.

mrot_t : Table des masses pour l'interpolation du moment cinétique au pas temporel précédent, réinitialisé dans `update.f`.

m23 : Table de masses simplifiant l'interpolation inverse, calculé dans `resout.f`.

m23_t : Table de masses simplifiant l'interpolation inverse au pas temporel précédent, réinitialisé dans `update.f`.

q : Table des abscisses entières pour linterpolation des variables quasi-statiques, calculé dans `lim_zc.f`.

qt : Vecteur nodal des abscisses entières, calculé dans `lim_zc.f`.

q_t : Table des abscisses entières au pas temporel précédent, réinitialisé dans `update.f`.

qt_t : Vecteur nodal des abscisses entières au pas temporel précédent, réinitialisé dans `update.f`.

r2 : Table de rayons simplifiant l'interpolation inverse, calculé dans `.f`.

r2_t : Table de rayons simplifiant l'interpolation inverse au pas temporel précédent, réinitialisé dans `update.f`.

xl : Abscisses des limites pour l'équilibre quasi-statique, calculé dans `coll_qs.f`.

xt_ptm : Vecteur nodal pour la tabulation de la perte de masse, calculé dans `pertm.f`.

xt_tds : Vecteur nodal pour la tabulation de l'énergie graviphique, calculé dans `cesam.f`.

xt_tds_t : Vecteur nodal pour la tabulation de l'énergie graviphique au pas temporel précédent, réinitialisé dans `update.f`.

x_ptm : Table des masses pour la tabulation de la perte de masse, calculé dans `.f`.

x_tds : Table des masses pour la tabulation de l'énergie graviphique, calculé dans `.f`.

x_tds_t : Table des masses pour la tabulation de l'énergie graviphique au pas temporel précédent, réinitialisé dans `update.f`.

r_zc_conv : Rayons des limites des zones convectives, calculé dans `lim_zc.f`.

m_zc : Masses aux limites des zones convectives, calculé dans `lim_zc.f`.

m_zc_t : Masses aux limites des zones convectives au pas temporel précédent, réinitialisé dans `update.f`.

r_zc : Rayons des zones convectives, calculé dans `lim_zc.f`.

r_zc_t : Rayons des zones convectives au pas temporel précédent, réinitialisé dans `update.f`.

r_ov : Rayons des zones convectives avec overshoot, calculé dans `lim_zc.f`.

r_ov_t : Rayons des zones convectives avec overshoot au pas temporel précédent, réinitialisé dans `update.f`.

age : Age du modèle, calculé dans `cesam.f`.

c_iben : Constante de Iben , calculé dans `cesam.f`.

mstar : Masse du modèle, calculé dans `atm.f`.

mstar_t : Masse du modèle au pas temporel précédent, réinitialisé dans `update.f`.

mw_tot : Moment cinétique total, calculé dans `resout.f`.

mw_tot_t : Moment cinétique total au pas temporel précédent, réinitialisé dans `update.f`.

psi0 : Constante de répartition, calculé dans `resout.f`

rstar : Rayon de l'étoile, calculé dans `atm.f`.

wrot : Vitesse angulaire, calculé dans `resout.f`.

wrot_t : Vitesse angulaire au pas temporel précédent, réinitialisé dans `update.f`.

– Variables PUBLIC integer :

jlim : Indices des limites zones radiatives / zones convectives, calculé dans `lim_zc.f`.

jlim_t : Indices des limites zones radiatives / zones convectives au pas temporel précédent, réinitialisé dans `update.f`.

model_num : Numéro du modèle, réinitialisé dans `cesam.f`.

knot : Nombre de points du vecteur nodal pour l'interpolation des variables quasi-statiques, calculé dans `coll.f`.

knotc : Nombre de points du vecteur nodal pour l'interpolation de la composition chimique, calculé dans `evol.f` ou `resout_diff_chim`.

knotc_t : Nombre de points du vecteur nodal pour l'interpolation de la composition chimique au pas temporel précédent, réinitialisé dans `update.f`.

knot_ptm : Nombre de points du vecteur nodal pour l'interpolation de la perte de masse, calculé dans `pertm.f`.

knotr : Nombre de points du vecteur nodal pour l'interpolation du vecteur de mélange, calculé dans `diffus.f`.

knotr_t : Nombre de points du vecteur nodal pour l'interpolation du vecteur de mélange, au pas temporel précédent, réinitialisé dans `update.f`.

knot_t : Nombre de points du vecteur nodal pour l'interpolation des variables quasi-statiques au pas temporel précédent, réinitialisé dans `update.f`.

knot_tds : Nombre de points du vecteur nodal pour l'interpolation de l'énergie graviphique, calculé dans `cesam.f`.

knot_tds_t : Nombre de points du vecteur nodal pour l'interpolation de l'énergie graviphique au pas temporel précédent, réinitialisé dans `update.f`.

lim : Indices des limites zones radiatives / zones convectives, calculé dans `lim_zc.f`.

lim_t : Indices des limites zones radiatives / zones convectives au pas temporel précédent, réinitialisé dans `update.f`.

model_num : Indice du dernier modèle calculé depuis le début de l'évolution compte tenu des reprises, réinitialisé dans `cesam`.

nb_modeles : Nombre de modèles calculés au cours du présent run, réinitialisé dans `cesam`.

nc_fixe : Nombre de points de la grille fixe d'interpolation de la composition chimique, calculé dans `evol.f`.

n_ch : Nombre de points de la grille d'interpolation de la composition chimique, calculé dans `evol.f`.

n_ch_t : Nombre de points de la grille d'interpolation de la composition chimique au pas temporel précédent, réinitialisé dans `update.f`.

n_ptm : Nombre de points de la grille d'interpolation de la perte de masse, calculé dans `pertm.f`.

n_qs : Nombre de points de la grille d'interpolation de l'équilibre quasi-statique, calculé dans `lim_zc.f`.

n_qs_t : Nombre de points de la grille d'interpolation de l'équilibre quasi-statique au pas temporel précédent, réinitialisé dans `update.f`.

n_rot : Nombre de points de la grille d'interpolation de la rotation, calculé dans `diffus.f`.

n_rot_t : Nombre de points de la grille d'interpolation de la rotation au pas temporel précédent, réinitialisé dans `update.f`.

n_tds : Nombre de points de la grille d'interpolation de l'énergie graviphique, calculé dans `.f`.

n_tds_t : Nombre de points de la grille d'interpolation de l'énergie graviphique au pas temporel précédent, réinitialisé dans `update.f`.

– Variables PUBLIC logical :

lconv : Début d'une zone convective, calculé dans `lim_zc.f`.

lconv_t : Début d'une zone convective au pas temporel précédent , réinitialisé dans `update.f`.

lhe_stop : Arrêt lorsque le cœur d'hélium a atteint la masse requise, calculé dans `cesam.f`.

lt_stop : Arrêt lorsque la température centrale a atteint la valeur requise, calculé dans `.f`.

lx_stop : Arrêt lorsque l'abondance centrale d'hydrogène a atteint la valeur requise, calculé dans `cesam.f`.

tot_conv : Le modèle est totalement convectif, calculé dans `lim_zc.f`.

tot_rad : Le modèle est totalement radiatif, calculé dans `.f`.

– Routines PUBLIC :

chim_gram, inter, sortie.

E.5 Module mod_etat

Le module `mod_etat` regroupe les routines concernées par l'équation d'état.
– Routines PRIVATE :

etat_ceff, etat_eff, etat_gong1, etat_gong2, etat_mhd, etat_opal, etat_opalX, etat_opalZ.
– Routines PUBLIC :

df_rotx, etat, saha.

E.6 Module mod_opa

Le module `mod_opa` regroupe les routines concernant le calcul de l'opacité.
– Routines PRIVATE :

kappa_cond, opa_gong, opa_houdek9, opa_int_zsx, opa_opalCO, opa_opal2, opa_yveline, opa_yveline_lisse.
– Routine PUBLIC :

opa.

E.7 Module mod_conv

Le module `mod_conv` regroupe les routines concernant la convection.
– Routines PRIVATE :

conv_a0, conv_cgm_reza, conv_cm, conv_cml, conv_cm_reza, conv_jmj.
– Routine PUBLIC :

conv.

E.8 Module mod_atm

Le module `mod_atm` regroupe les routines de restitution de l'atmosphère.

- Variables PUBLIC double précision :

  bp_atm : Tableau des coefficients des B-splines d'interpolation de l'atmosphère, calculé dans `coll_atm.f`.

  bp_atm_t : Tableau des coefficients des B-splines d'interpolation de l'atmosphère au pas temporel précédent, réinitialisé dans `update.f`.

  dlpp_atm : Tableau des quotients $d\ln P_{\rm gaz}/d\ln P_{\rm tot}$, calculé dans `atm.f`.

  m_atm : Tableau des masses, abscisses lagrangiennes ,calculé dans `coll_atm.f`.

  p_atm : Tableau des pressions gazeuses, calculé dans `atm.f`.

  pt_atm : Tableau des pressions totales, calculé dans `atm.f`.

  r_atm : Tableau des rayons , calculé dans `atm.f`.

  tau : Tableau des profondeurs optiques, calculé dans `atm.f`.

  t_atm : Tableau des températures, calculé dans `atm.f`.

  x_atm : Tableau des abscisses pour la restitution de l'atmosphère, calculé dans `atm.f`.

  x_atm_t : Tableau des abscisses pour la restitution de l'atmosphère au pas temporel précédent, réinitialisé dans `update.f`.

  xt_atm : Vecteur nodal des abscisses pour la restitution de l'atmosphère, calculé dans `lim_atm.f`.

  tau_min : Epaisseur optique de la couche externe de l'atmosphère, initialisé dans `tdetau.f`.

- Variables PRIVATE double précision :

  delfim : Pente de la fonction de répartition à gauche de la couche d'indice `n23_atm`, calculé dans `lim_atm.f`.

  delfip : Pente de la fonction de répartition à droite de la couche d'indice `n23_atm`, calculé dans `lim_atm.f`.

  ltaue : $\ln(\tau_{\rm min})$, initialisé dans `lim_atm.f`.

  ltauf : $\ln(\tau_{\rm max})$, initialisé dans `lim_atm.f`.

  tau_min : Epaisseur optique de la couche externe de l'atmosphère, initialisé dans `tdetau.f`.

- Variables PRIVATE integer :

  Ipgt : Indice de la variable pression gazeuse, initialisé dans `lim_atm.f`.

  ne_atm : Nombre d'équations pour la restitution de l'atmosphère, initialisé dans `lim_atm.f`.

  n23_atm : Indice de la couche définissant le rayon bolométrique où $T(\tau*) = T_{\rm eff}$, initialisé dans `lim_atm.f`.

- Variables PRIVATE logical :

rad : La loi T(tau) est purement radiative, initialisé dans `tdetau.f`.

– Variable PRIVATE character :

nom_atm : Nom de la routine de loi $T(\tau)$ à utiliser, lu dans `lit_nl.f`.

– Routines PRIVATE :

coll_atm, edding, eq_atm, hopf, k5750, k5777, lim_atm, lim_gong1, lim_tau1, roger00, roger02, roger05, roger10a, taueff, trho

– Routines PUBLIC :

atm, tdetau, thermo_atm.

E.9 Module mod_nuc

Le module `mod_nuc` regroupe les routines concernant les réactions thermonucléaires.

– Paramètres PRIVATE integer :

nelem_ini : Nombre maximum d'éléments.

niso_tot : Nombre maximum d'isotopes.

nreac_tot : Nombre maximum de réactions thermonucléaires.

– Variables PRIVATE double précision :

ab : Tableau des abondances, initialisé dans `abon_ini.f`.

abon_rela : Tableau des abondances relatives métal/Z, initialisé dans `abon_ini.f`.

m : Masses atomiques, initialisé dans `abon_ini.f`.

c : Charges des éléments chimiques, initialisé dans `abon_ini.f`.

be7sbe9 : Rapport isotopique ^{7}Be/^{9}Be, initialisé dans `abon_ini.f`.

be7sz : Rapport isotopique ^{7}Be/Z, initialisé dans `abon_ini.f`.

c13sc12 : Rapport isotopique ^{13}C/^{12}C, initialisé dans `abon_ini.f`.

h2sh1 : Rapport isotopique ^{2}H/^{1}H, initialisé dans `abon_ini.f`.

he3she4 : Rapport isotopique ^{3}He/^{4}He, initialisé dans `abon_ini.f` .

he3she4z :Rapport isotopique ^{3}He/^{4}He, avec ^{2}H primordial transformé en ^{3}He, initialisé dans `abon_ini.f`.

li6sli7 : Rapport isotopique ^{6}Li/^{7}Li, initialisé dans `abon_ini.f`.

mg25smg24 : Rapport isotopique ^{25}Mg/^{24}Mg, initialisé dans `abon_ini.f`.

mg26smg25 :Rapport isotopique ^{26}Mg/^{25}Mg, initialisé dans `abon_ini.f`.

ne22sne20 : Rapport isotopique ^{22}Ne/^{20}Ne, initialisé dans `abon_ini.f`.

n15sn14 : Rapport isotopique ^{15}N/^{14}N, initialisé dans `abon_ini.f`.

o17so16 : Rapport isotopique ^{17}O/^{16}O, initialisé dans `abon_ini.f`.

o18so16 : Rapport isotopique ^{18}O/^{16}O, initialisé dans `abon_ini.f`.

t_sup : Température maximale de tabulation des réactions thermonucléaires, initialisé dans `tabul_nuc.f`.

– Variables PRIVATE integer :

izz : Table des charges des noyaux interagissant dans les réactions thermonucléaires, initialisé dans `tabul_nuc.f`.

nreac : Nombre de réactions thermonucléaires du réseau utilisé, initialisé dans `tabul_nuc.f`.

– Variable PRIVATE character :

elem : Noms des éléments chimiques, initialisé dans `abon_ini.f`.

– Routines PRIVATE :

abon_ini, iben, pp1, pp3, ppcno10BeBFe, ppcno10Fe, pcno10K, ppcno10, ppcno11, ppcno12Be, ppcno12BeBFe, ppcno12Li, ppcno12, ppcno3a12Ne, ppcno3a9, ppcno3ac10, ppcno9, ppcno9Fe, rq_reac, tabul_nuc, taux_nuc.

– Routines PUBLIC :

nuc, vent.

E.10 Module mod_bp_for_alecian

Le module `mod_bp_for_alecian` regroupe les routines concernant le calcul des accélérations radiatives suivant le second formalisme de G.Alécian. Développé par B.Pichon, il constitue essentiellement une interface entre les routines fournies par G.Alécian et CESAM2*k*. D'origine externe, les fonctions des composantes de ce module ne sont pas détaillées.

E.11 Module mod_evol

Le module `mod_evol` regroupe les routines concernant l'évolution temporelle avec et sans diffusion de la composition chimique et du moment cinétique.

– Paramètres PRIVATE double précision :

eps : Petit écart permettant l'estimation numérique de dérivées.

un_eps : Ecart pour l'estimation numérique de dérivées.

– Paramètres PRIVATE integer :

m_mix : Ordre des B-splines d'interpolation du vecteur de mélange.

nmix : Nombre d'inconnues d'interpolation du vecteur de mélange.

nrl : Nombre d'inconnues d'interpolation pour les coefficients de la diffusion du moment cinétique.

– Variables PRIVATE double précision :

coef_rl : Tableau des coefficients des B-splines d'interpolation des coefficients de la diffusion du moment cinétique, calculé dans `coeff_rota.f`.

fmix : Tableau des coefficients des B-splines d'interpolation du vecteur de mélange, calculé dans `diffus.f`.

mrl : Abscisses d'interpolation des coefficients de la diffusion du moment cinétique, réinitialisé dans `coeff_rota.f`.

mrlt : Vecteur nodal d'interpolation des coefficients de la diffusion du moment cinétique, réinitialisé dans `coeff_rota.f`.

x_mix : Abscisses d'interpolation du vecteur de mélange, calculé dans `diffus.f`.

x_mixt : Vecteur nodal d'interpolation du vecteur de mélange, calculé dans `diffus.f`.

mlim_vent : Masse au dessous de laquelle le vent n'est plus actif, initialisé dans `evol.f`.

mlim_w : Masse au dessous de laquelle la perte de moment cinétique n'est plus active, calculé dans `pertw.f`.

– Variable PUBLIC double précision :

nu_min : Pseudo valeur centrale de la masse pour éviter les singularités, calculé dans `coeff_rota.f`.

– Variables PRIVATE integer :

knotrl : Dimension du vecteur nodal d'interpolation des coefficients de la diffusion du moment cinétique, calculé dans `coeff_rota.f`.

knot_mix : Dimension du vecteur nodal d'interpolation du vecteur de mélange, calculé dans `diffus.f`.

n_rl : Nombre de points d'interpolation des coefficients de la diffusion du moment cinétique, réinitialisé dans `coeff_rota.f`.

n_mix : Nombre de points d'interpolation du vecteur de mélange, réinitialisé dans `diffus.f`.

– Variables PUBLIC logical :

ini_rot : , calculé dans `.f`.

– Routines PRIVATE :

alecian1, coeff_rota, collision, coulomb, diffm, diffm_br, diffm_mp, difft, difft_nu, diffus, diffw, diffw_mpz, eq_diff_chim, eq_diff_rota, f_rad, pertw, pertw_loc, pertw_ptm, pertw_sch, resout_chim, resout_rota, rk_imps.

– Routines PUBLIC :

evol.

E.12 Module mod_static

Le module `mod_static` regroupe les routines concernées par la résolution des équations de l'équilibre quasi-statique.

– Paramètre PRIVATE character :

nom_qs : Noms des variables quasi-statiques.

– Variables PRIVATE double précision :

fac : Facteurs de répartition, calculé dans `lim_zc.f`.

xcoll : Abscisses des points de collocation, réinitialisé dans `coll_qs.f`.

– Routines PRIVATE :

coll_qs, dgrad, lim_zc, pertm_ext, pertm_msol, pertm_tot, pertm_waldron,static, static_m, static_r, update.

– Routines PUBLIC :

pertm, resout, thermo.

E.13 Module mod_cesam

Le module `mod_cesam` regroupe les routines concernant la gestion générale du calcul.

– Routines PRIVATE :

add_ascii, ascii, des, des_m, des_r, dnunl, initialise_rota, list, output, osc_adia, osc_invers, osc_nadia, u_initial, w_initial.

– Routine PUBLIC :

cesam.

E.14 Module mod_exploit

Le module `mod_exploit` regroupe des routines concernées par l'exploitation des résultats.

– Routine PRIVATE :

ascii.

– Routines PUBLIC :

add_ascii, ctes_85, ctes_94, ini_ctes, inter_atm, lit_binaire, lit_hr, lit_nl, min_max, min_max_cond, osc_adia, osc_invers, osc_nadia, output, read_ascii, write_nl.

Annexe F

Contenu des sous-directory

F.1 Sous-directory SOURCE

abon_ini.f : Initialisation des abondances, routine PRIVATE du module `mod_nuc`.

add_ascii.f : Complète l'ensemble des variables pour sorties ASCII, routine PRIVATE du module `mod_cesam`.

alecian1.f : Calcul des accélérations radiatives, routine PRIVATE du module `mod_evol`.

arb_rom.f : Transforme la notation arabe en romaine, fonction PUBLIC du module `mod_numerique`.

ascii.f : Création d'un fichier de sortie ASCII personnalisé, routine PRIVATE du module `mod_cesam`.

atm.f : Routine générique de restitution de l'atmosphère, routine PUBLIC du module `mod_atm`.

boite.f : Dessine une boite de dimensions $\pm dx \times dy$ autour du point de coordonnées (x, y), routine PUBLIC du module `mod_numerique`.

box.f : Dessine une boite asymétrique autour du point de coordonnées (x, y), routine PUBLIC du module `mod_numerique`.

bsp1ddn.f : Interpolation 1D avec une spline polynômiale + dérivées, routine PUBLIC du module `mod_numerique`.

bsp1dn.f : Interpolation 1D avec une spline polynômiale, routine PUBLIC du module `mod_numerique`.

bsp_dis.f : Interpolation 1D avec discontinuités avec une spline polynômiale, routine PUBLIC du module `mod_numerique`.

bval0.f : Calcul les B-splines normalisées en un point de leur domaine, routine PRIVATE du module `mod_numerique`.

bval1.f : Calcul les B-splines normalisées et de leurs dérivées 1-ières en un point de leur domaine, routine PUBLIC du module `mod_numerique`.

bvald.f : Calcul les B-splines normalisées et de leurs dérivées en un point de leur domaine, routine PUBLIC du module `mod_numerique`.

cesam.f : Gestion de l'évolution, constitue en fait le programme principal, routine PUBLIC du module `mod_cesam`.

cesam2k.f : Programme principal pour exploitation.

cesam2k-dbg.f : Programme principal pour debug.

cesam_T.f : Programme principal pour exploitation de tests.

cesam_T-dbg.f : Programme principal pour debug de tests.

chim_gram.f : Transforme les abondances par mole en abondances par gramme, routine PUBLIC du module `mod_variables`.

coll.f : Détermination des abscisses des points de collocation, routine PUBLIC du module `mod_numerique`.

col_atm.f : Résolution des équations de l'atmosphère, routine PRIVATE du module `mod_atm`.

col_qs.f : Résolution des équations de l'équilibre quasi-statique, routine PRIVATE du module `mod_static`.

collision.f : Calcul des intégrales de collision, routine PRIVATE du module `mod_evol`.

colpnt.f : Initialisation des abscisses des points de collocation, routine PRIVATE du module `mod_numerique`.

conv.f : Routine générique du calcul de la convection, routine PUBLIC du module `mod_conv`.

conv_a0.f : Calcul de la convection MLT avec longueur de mélange nulle aux limites ZR/ZC, routine PRIVATE du module `mod_conv`.

conv_cgm_reza.f : Calcul de la convection selon Canuto & Mazitelli avec la modification CGM, routine PRIVATE du module `mod_conv`.

conv_cm.f : Calcul de la convection selon Canuto & Mazitelli, routine PRIVATE du module `mod_conv`.

conv_cm_reza.f : Calcul de la convection selon Canuto & Mazitelli avec la modification de R.Samadi, routine PRIVATE du module `mod_conv`.

conv_cml.f : Calcul de la convection selon Canuto & Mazitelli avec la longueur de mélange égale à la plus courte distance des bords de la zone convective, routine PRIVATE du module `mod_conv`.

conv_jmj.f : Calcul de la convection MLT, routine PRIVATE du module `mod_conv`.

coulomb.f : Calcul du logarithme de Coulomb, routine PRIVATE du module `mod_evol`.

ctes_85.f : Initialisation de constantes physiques, routine PRIVATE du module `mod_donnees`.

ctes_94.f : Initialisation de constantes physiques, routine PRIVATE du module `mod_donnees`.

delete_doubles : Elimination des éléments identiques dans un tableau ordonné. Routine PUBLIC du module `mod_numerique`.

des.f : Routine générique du dessin au cours de l'évolution, routine PRIVATE du module `mod_cesam`.

des_m.f : Dessin en masse au cours de l'évolution, routine PRIVATE du module `mod_cesam`.

des_r.f : Dessin en rayon au cours de l'évolution, routine PRIVATE du module `mod_cesam`.

df_rotx.f : transformation des dérivées partielles $(P,T,X) \rightarrow (\rho,T,X)$, routine PUBLIC du module `mod_etat`.

dgrad.f : Calcul de la différence des gradients, routine PUBLIC du module `mod_variables`.

difdiv.f : Calcul des différences divisées, routine PRIVATE du module `mod_numerique`.

diffm.f : Routine générique du calcul des coefficients de diffusion microscopique, routine PRIVATE du module `mod_evol`.

diffm_br.f : Calcul des coefficients de diffusion microscopique selon Burgers, routine PRIVATE du module `mod_evol`.

diffm_mp.f : Calcul des coefficients de diffusion microscopique selon Michaud & Proffit, routine PRIVATE du module `mod_evol`.

difft.f : Routine générique du calcul des coefficients de diffusion turbulente, routine PRIVATE du module `mod_evol`.

difft_nu.f : Calcul des coefficients de diffusion turbulente et radiative, routine PRIVATE du module `mod_evol`.

diffus.f : Gestion des équations de diffusion des espèces chimiques et du moment cinétique, routine PRIVATE du module `mod_evol`.

diffw.f : Routine générique du calcul des coefficients de diffusion du moment cinétique, routine PRIVATE du module `mod_evol`.

diffw_mpz.f : Calcul des coefficients de diffusion du moment cinétique, selon Mathis, Palacios & Zahn [2004], routine PRIVATE du module `mod_evol`.

dnunl.f : Calcul approximatif de la grande séparation, routine PRIVATE du module `mod_cesam`.

ecrit_ascii : Formation et écriture des fichiers ASCII, routine subordonnée de `cesam.f`.

edding.f : Loi $T(\tau)$ d'Eddington, routine PRIVATE du module `mod_atm`.

eq_atm.f : Calcul des coefficients des équations de restitution de l'atmosphère, routine PRIVATE du module `mod_atm`.

eq_diff_chim.f : Calcul des coefficients des équations de diffusion de la composition chimique, routine PRIVATE du module `mod_evol`.

eq_diff_rota.f : Calcul des coefficients des équations de diffusion du moment cinétique, routine PRIVATE du module `mod_evol`.

etat.f : Routine générique du calcul de l'équation d'état, routine PUBLIC du module `mod_etat`.

etat_ceff.f : Equation d'état CEFF, routine PRIVATE du module `mod_etat`.

etat_eff.f : Equation d'état EFF, routine PRIVATE du module `mod_etat`.

etat_gong1.f : Equation d'état GONG1, routine PRIVATE du module `mod_etat`.

etat_gong2.f : Equation d'état GONG2, routine PRIVATE du module `mod_etat`.

etat_mhd.f : Equation d'état MHD, routine PRIVATE du module `mod_etat`.

etat_opal.f : Equation d'état OPAL, routine PRIVATE du module `mod_etat`.

etat_opalX.f : Equation d'état OPAL utilisable pour $Z > 0.1$, routine PRIVATE du module `mod_etat`, *cf.* § 8.54 (Page 166).

etat_opalZ.f : Equation d'état OPAL utilisable pour $Z > 0.1$, routine PRIVATE du module `mod_etat`, *cf.* § 8.54 (Page 166).

evol.f : Gestion de l'évolution temporelle de la composition chimique, routine PUBLIC du module `mod_evol`.

fcmax : Détermine la nécessité d'utiliser le nombre maximum de couche lors du calcul des derniers modèles. Routine subordonnée de `resout` du module `mod_static`.

fermi_dirac.f : Intégrales de Fermi-Dirac, routine PUBLIC du module `mod_numerique`.

from_alecian.f : package de calcul des accélérations radiatives Alécian2, routine PRIVATE du module `mod_bp_for_alecian`.

f_rad.f : Routine générique du calcul des acclérations radiatives, routine PRIVATE du module `mod_evol`.

gauss_band.f : Résolution d'un système linéaire bande, routine PUBLIC du module `mod_numerique`.

genere_bases.f : Génération de bases de B-splines discontinues et continues, routine PUBLIC du module `mod_numerique`.

hopf.f : Loi $T(\tau)$ de Hopf, routine PRIVATE du module `mod_atm`.

horner.f : Algorithme de Horner , routine PRIVATE du module `mod_numerique`.

iben.f : Calcul de l'énergie de contraction selon Iben, routine PRIVATE du module `mod_nuc`.

ini_ctes.f : Routine générique d'initialisation des constantes physiques, routine PUBLIC du module `mod_donnees`.

initialise_rota.f : Initialisation des variables de la diffusion du moment cinétique, routine PRIVATE du module `mod_cesam`.

inter.f : Interpolation inverse des variables, routine PUBLIC du module `mod_variables`.

inter_atm.f : Interpolation inverse pour l'atmosphère, routine PUBLIC du module `mod_exploit`.

intgauss.f : Initialisation des poids et des abscisses pour l'intégration de Gauss, routine PUBLIC du module `mod_numerique`.

journal : Description des aménagements dans CESAM2*k*.

k5750.f : Loi $T(\tau)$ de Kurucz, $T_{\rm eff} = 5750$K, routine PRIVATE du module `mod_atm`.

k5777.f : Loi $T(\tau)$ de Kurucz, $T_{\rm eff} = 5777$K, routine PRIVATE du module `mod_atm`.

kappa_cond.f : opacités conductives, routine PRIVATE du module `mod_opa`.

lim_atm.f : Gestion de la restitution de l'atmosphère, routine PRIVATE du module `mod_atm`.

lim_gong1.f : Atmosphère monocouche, cas GONG1, routine PRIVATE du module `mod_atm`.

lim_tau1.f : Atmosphère monocouche, routine PRIVATE du module `mod_atm`.

lim_zc.f : Gestion des limites ZR/ZC, routine PRIVATE du module `mod_static`.

linf.f : Recherche de l'encadrement d'une valeur, routine PUBLIC du module `mod_numerique`.

list.f : Listing des modèles, routine PRIVATE du module `mod_cesam`.

list_cesam : Liste des modules à compiler.

lit_binaire.f : Lecture d'un fichier binaire de type `mon_modele_B.*`.

lit_nl.f : Lecture des NAMELISTs, routine PUBLIC du module `mod_donnees`.

lit_hr.f : Lecture des fichiers de type `mon_modele.HR`, routine PUBLIC du module `mod_exploit`.

matinv.f : Inversion de matrice, routine PUBLIC du module `mod_numerique`.

min_max.f : Maximum/minimum d'une table, routine PUBLIC du module `mod_numerique`.

mod_atm.f : module regroupant les variables et les routines de résolution des équations différentielles pour la restitution de l'atmosphère.

mod_bp_for_alecian.f : module regroupant les routines de calcul de l'accélération radiative selon G.Alécian.

mod_cesam.f : module contenant ce qui constitue, de fait, le programme principal *i.e.* la routine `cesam` et les routines de gestion.

mod_conv.f : module regroupant les routines de convection.

mod_donnees.f : module contenant les données du modèle, les constantes physiques et leurs routines d'initialisation.

mod_etat.f : module regroupant les équations d'état.

mod_evol.f : module regroupant les routines relatives à l'évolution de la composition chimique.

mod_exploit.f : module regroupant des routines et programmes pour l'exploitation des modèles.

mod_kind.f : module regroupant les définitions de types de variables.

mod_nuc.f : module regroupant les routines des réactions nucléaires et d'initialisation de la mixture.

mod_numerique.f : module regroupant les outils numériques et informatiques.

mod_opa.f : module regroupant les routines d'opacité.

mod_static.f : module regroupant les variables et les routines de résolution des équations différentielles pour l'équilibre quasi-statique.

mod_variables.f : module regroupant les variables et quelques fonction de calcul de quantités dérivées des variables.

modif_mix.f : Routine permettant la personnalisation d'une mixture, routine PRIVATE du module `mod_nuc`.

neville.f : Algorithme de Neville, routine PUBLIC du module `mod_numerique`.

newspl.f : Changement de base de spline, routine PUBLIC du module `mod_numerique`.

newton.f : formule d'interpolation de Newton, routine PUBLIC du module `mod_numerique`.

noedif.f : Création de la base pour la résolution des équations différentielles selon de Boor, routine PUBLIC du module `mod_numerique`.

noein.f : Création de la base d'interpolation, routine du module `mod_numerique`.

noeu_dis.f : Création du vecteur nodal pour interpolation avec discontinuités, routine PRIVATE du module `mod_numerique`.

noeud.f : Création d'un vecteur nodal à partir d'un vecteur de mutiplicité, routine PUBLIC du module `mod_numerique`.

nuc.f : Routine générique de calcul des réactions nucléaires, routine PUBLIC du module `mod_nuc`.

opa.f : Routine générique du calcul d'opacité, routine PUBLIC du module `mod_opa`.

opa_gong.f : opacité de GONG, routine PRIVATE du module `mod_opa`.

opa_houdek9.f : opacité OPAL + Alexander, interpolation de Houdek, routine PRIVATE du module `mod_opa`.

opa_int_zsx.f : opacité OPAL, routine PRIVATE du module `mod_opa`.

opa_opal2.f : opacité OPAL permettant de dépasser $Z > 0.1$, routine PRIVATE du module `mod_opa`, *cf.* § 8.81 (Page 184).

opa_opalCO.f : opacité OPAL permettant de dépasser $Z > 0.1$, routine PRIVATE du module `mod_opa`, *cf.* § 8.81 (Page 184).

opa_yveline.f : opacité OPAL + Alexander, interpolation de Yveline, routine PRIVATE du module `mod_opa`.

opa_yveline_lisse.f : opacité OPAL + Alexander, raccord d'Yveline, routine PRIVATE du module `mod_opa`.

osc_adia.f : formation du fichier d'oscillations adiabatiques, routine PRIVATE du module `mod_cesam`.

osc_invers.f : formation du fichier ASCII pour inversions, routine PRIVATE du module `mod_cesam`.

osc_nadia.f : formation du fichier d'oscillations non-adiabatiques, routine PRIVATE du module `mod_cesam`.

output.f : Routine générique de création des fichiers de sortie en ASCII, routine PRIVATE du module `mod_cesam`.

pause.f : pause avec commentaire, routine PUBLIC du module `mod_numerique`.

pertm.f : Routine générique du calcul de la perte de masse, routine PRIVATE du module `mod_static`.

pertm_ext.f : perte de masse linéaire, routine PRIVATE du module `mod_static`.

pertm_msol.f : perte de masse linéaire limitée, routine PRIVATE du module `mod_static`.

pertm_tot.f : perte de masse linéaire + $E = mc^2$, routine PRIVATE du module `mod_static`.

pertm_waldron.f : perte de masse selon Waldron, routine PRIVATE du module `mod_static`.

pertw.f : Routine générique de perte de moment cinétique, routine PRIVATE du module `mod_evol`.

pertw_loc.f : perte locale de moment cinétique, routine PRIVATE du module `mod_evol`.

pertw_ptm.f : perte de moment cinétique associée à une perte de masse, routine PRIVATE du module `mod_evol`.

pertw_sch.f : perte de moment cinétique selon Schumanich, routine PRIVATE du module `mod_evol`.

polyder.f : Algorithme de Horner, routine PUBLIC du module `mod_numerique`.

pp1.f : Cycle PP simplifié, routine PRIVATE du module `mod_chim`.

pp3.f : Cycle PP, routine PRIVATE du module `mod_chim`.

ppcno10.f : Cycles PP + CNO 10 éléments, routine PRIVATE du module `mod_chim`.

ppcno10BeBFe.f : Cycles PP + CNO 10 éléments + Fe, routine PRIVATE du module `mod_nuc`.

ppcno10Fe.f : Cycles PP + CNO 10 éléments + Fe, routine PRIVATE du module `mod_nuc`.

ppcno10K.f : Cycles PP + CNO 10 éléments + K, routine PRIVATE du module `mod_nuc`.

ppcno11.f : Cycles PP + CNO 11 éléments, routine PRIVATE du module `mod_nuc`.

ppcno12.f : Cycles PP + CNO 12 éléments, routine PRIVATE du module `mod_nuc`.

ppcno12Be.f : Cycles PP + CNO 12 éléments + Be, routine PRIVATE du module `mod_nuc`.

ppcno12BeBFe.f : Cycles PP + CNO 12 éléments + Be, routine PRIVATE du module `mod_nuc`.

ppcno12Li.f : Cycles PP + CNO 12 éléments + Li, routine PRIVATE du module `mod_nuc`.

ppcno3a12Ne.f : Cycles PP + CNO + 3α 12 éléments, routine PRIVATE du module `mod_nuc`.

ppcno3a9.f : Cycles PP + CNO + 3α 9 éléments, routine PRIVATE du module `mod_nuc`.

ppcno3ac10.f : cycles PP + CNO + 3α + C + O 10 éléments, routine PRIVATE du module `mod_nuc`.

ppcno9.f : Cycles PP + CNO 9 éléments, routine PRIVATE du module `mod_nuc`.

ppcno9Fe.f : Cycles PP + CNO 9 éléments, routine PRIVATE du module `mod_nuc`.

print_ctes.f : Ecriture des constantes, routine PUBLIC du module `mod_donnees`.

read_ascii.f : Lecture du fichier ASCII, routine du module `mod_exploit`.

resout.f : Gestion de la résolution de l'équilibre quasi-statique + évolution temporelle, routine du module `mod_static`.

rk_imps.f : Algorithme Runge-Kutta implicite, routine PRIVATE du module `mod_evol`.

roger00.f : Loi $T(\tau)$ Kurucz Fe/H=0, routine PRIVATE du module `mod_atm`.

roger02.f : Loi $T(\tau)$ Kurucz Fe/H=-0.2, routine PRIVATE du module `mod_quasi_static`.

roger05.f : Loi $T(\tau)$ Kurucz Fe/H=-0.5, routine PRIVATE du module `mod_atm`.

roger10a.f : Loi $T(\tau)$ Kurucz Fe/H=-1.0 α-enhanced, routine PRIVATE du module `mod_atm`.

rq_reac.f : Calcul des énergies des réactions nucléaires, routine PRIVATE du module `mod_nuc`.

saha.f : Equation de Saha, routine PUBLIC du module `mod_etat`.

schu58_n.f : Algorithme de Schumaker, routine PRIVATE du module `mod_numerique`.

shell.f : Routine de tri, routine PUBLIC du module `mod_numerique`.

static.f : Routine générique du calcul des coefficients de l'équilibre quasi-statique, routine PRIVATE du module `mod_static`.

static_m.f : Calcul des coefficients de l'équilibre quasi-statique, cas lagrangien, routine PRIVATE du module `mod_quasi_static`.

static_r.f : Calcul des coefficients de l'équilibre quasi-statique, cas eulérien, routine PRIVATE du module `mod_quasi_static`.

sum_n.f : Intégration d'une spline, routine PUBLIC du module `mod_numerique`.

tabul_nuc.f : tabulation des réactions thermonucléaires, routine PRIVATE du module `mod_nuc`.

taueff.f : pour une loi $T(\tau)$, détermination de l'épaisseur optique où $T = T_{\rm eff}$, routine PRIVATE du module `mod_atm`.

taux_nuc.f : taux des réactions thermonucléaires, routine PRIVATE du module `mod_nuc`.

tdetau.f : Routine générique de loi $T(\tau)$, routine PUBLIC du module `mod_atm`.

thermo.f : Calcul des variables thermodynamiques pour l'équilibre quasi-statique, routine PUBLIC du module `mod_static`.

thermo_atm.f : Calcul des variables thermodynamiques pour l'atmosphère, routine PUBLIC du module `mod_atm`.

trho.f : Interpolation en T et ρ pour lois $T(\tau)$ de Kurucz, routine PRIVATE du module `mod_atm`.

u_initial.f : Initialisation de la vitesse de circulation méridienne, routine PRIVATE du module `mod_cesam`.

update.f : passage des variables du temps t+dt au temps t, fonction PRIVATE du module `mod_static`.

vent.f : Gestion de la composition chimique du vent, routine PRIVATE du module `mod_nuc`.

write_nl.f : Ecriture des NAMELISTs, routine PRIVATE du module `mod_cesam`.

w_initial.f : Initialisation de la vitesse angulaire, fonction PRIVATE du module `mod_cesam`.

zoning.f : Répartition équidistante d'une fonction croissante, routine PUBLIC du module `mod_numerique`.

z14xcotrin21.f : Package d'interpolation d'opacité, n'est pas inclus dans un module.

F.2 Sous-directory EXPLOIT

2d-2.pms : Fichier ASCII d'initialisation de pré-séquence principale, on obtient une température centrale de l'ordre de 100 000K.

5d-4.pms : Fichier ASCII d'initialisation de pré-séquence principale, on obtient une température centrale de l'ordre de 500 000K.

8d-5.pms : Fichier ASCII d'initialisation de pré-séquence principale, on obtient une température centrale de l'ordre de 1 000 000K.

calib2k.f : Programme de calibration.

des2k_ZC.f : Programme de dessin des zones convectives.

des2k_abon.f : Programme de dessin des abondances.

des2k_abontc.f : Programme de dessin de X, Y et Z au centre, en fonction du temps.

des2k_abonts.f : Programme de dessin de X, Y et Z en surface, en fonction du temps.

des2k_bin.f : Programme de dessin des variables quasi-statiques en fonction de la masse ou du rayon à partir d'un fichier binaire.

des2k_dhve.f : Programme de dessin des variables de la diffusion du moment cinétique à partir d'un fichier binaire de type `mon_modele_B.*`.

des2k_hr.f : Programme de dessin de diagrammes HR.

des2k_opa.f : Programme de dessin des différences relatives de deux sources d'opacité, le long d'un modèle.

des2k_osc.f : Programme de dessin des principales variables d'un fichier d'oscillations de type `mon_modele-ad.osc`.

des2k_rot.f : Programme des dessins séparés des variables de la diffusion du moment cinétique à partir du fichier binaire optionnel créé dans la routine `coeff_rota.f`.

des2k_rot_ext.f : Programme de dessin, en fonction du temps, de la vitesse de rotation externe en Km/s ou en radian/s à partir d'un fichier HR.

des2k_vaiss.f : Programme de dessin de la fréquence de Brunt-Väissälä, à partir d'un fichier binaire de type `mon_modele_B.*`.

device : Exemple de ficheir de personnalisation des paramètres de dessin *"on line"*, *cf.* § 4 (Page 37).

f037_2k.f : Formation d'un fichier pour le calcul des oscillations à partir d'un fichier binaire de type `mon_modèle_B.dat`.

fichier_vent.f : Programme de contruction, pour le vent, d'un fichier de composition chimique différente de celle des couches externes.

langue : Exemple de fichier de langue.

m010.zams : Fichier ASCII d'initialisation de séquence principale d'âge zéro homogène, pour $1M_{\odot}$.

m020.zams : Fichier ASCII d'initialisation de séquence principale d'âge zéro homogène, pour $2M_{\odot}$.

m050.zams : Fichier ASCII d'initialisation de séquence principale d'âge zéro homogène, pour $5M_{\odot}$.

makefile : Exemple de `MAKEFILE` pour la formation du module exécutable, *cf.* § 1.6.1 (Page 5).

mixture : Exemple de fichier de mixture initiale, différente de celles implémentées dans CESAM2k, *cf.* § 4.2 (Page 38).

modif_mix : Exemple de fichier de modification de mixture pour ***tous les modèles*** calculés dans l'environnement, *cf.* § 4.4 (Page 40).

mon_modele.don : Exemple de fichier de données.

mon_modele.modif_mix : Exemple de fichier de modification de mixture pour ***le*** modèle `mon_modele`.

mon_modele.rap_iso : Exemple de fichier de modification des rapports isotopiques pour ***le*** modèle `mon_modele`

planet : Exemple de fichier permettant de définir les paramètres des chutes de planètoïdes, *cf.* § 7.8.2 (Page 99).

physique45 : Exemple de fichier permettant de personnaliser la physique à utiliser dans le cas d'un emploi de fichier de données d'une version précédente de CESAM.

rap_iso : Exemple de fichier de modification des rapports isotopiques.

reglages : Exemple de fichier de réglage.

sortie_ascii : Exemple de fichier de personnalisation des sorties ASCII.

vent : Exemple de fichier de composition chimique du vent.

zoom : Exemple de fichier de personnalisation des dessins "on line".

F.3 Sous-directory TESTS

Certains de ces programmes de test ne sont peut être pas à jour de la toute dernière version de CESAM2k.

test_acc_rad.f : Programme de test des accélérations radiatives.

test_atm.f : Programme de test de l'intégration de l'atmosphère.

test_bsp_dis.f : Programme de test de l'interpolation par B-splines avec discontinuités.

test_cesam.f : Programme de test de CESAM2k.

test_coef_diff.f : Programme de test du calcul des coefficients de diffusion microscopique.

test_coeff_rota.f : Programme de test du calcul des coefficients de la diffusion du moment cinétique.

test_collision.f : Programme de test des coefficients de collision.

test_colloc.f : Programme de test de la construction d'un vecteur nodal pour la collocation.

test_convection.f : Programme de test d'une routine de convection.

test_data_ceff.f : Programme de test des DATA de l'équation d'état CEFF.

test_data_eff.f : Programme de test des DATA de l'équation d'état EFF.

test_der_rotx.f : Programme de test de transformation de dérivation.

test_etat.f : Programme de test d'équation d'état.

test_jacobien_reac_nuc.f : Programme de test du jacobien d'un réseau de réactions ucléaires.

test_lim_ZC.f : Programme de test de la localisation des zones convectives.

test_nl.f : Programme de test de lecture des NAMELISTs.

test_opa.f : Programme de test des opacités.

test_opa_opal2.f : Programme de test des opacités OPAL2.

test_opacite.f : Programme de test des opacités avec différentes mixtures.

test_read_osc.f : Programme de test de la lecture des fichiers ASCII d'oscillations.

test_rk_imps.f : Programme de test de la routine d'intégration des réactions du réseau nucléaire sans diffusion microscopique.

test_saha.f : Programme de test du calcul des taux d'ionisation.

test_tabul_reac.f : Programme de test de la tabulation des taux des réactions nucléaires.

test_tdetau.f : Programme de test d'un loi $T(\tau)$.

test_thermo.f : Programme de test du calcul des principales variables physiques de la structure interne.

test_thermo_atm.f : Programme de test du calcul des principales variables physiques de l'atmosphère.

F.4 Sous-directory SCRIPTS

calib2k_pms : Calibration d'un modèle solaire incluant la PMS.

calib2k_zams : Calibration d'un modèle solaire initialisé sur la ZAMS homogène.

compile2k : Compilation d'une routine avec le compilateur `lf95`.

compile2k-dbg : Compilation d'une routine avec le compilateur `lf95` et options de debug.

compile2k-dbg_list : Compilation d'une routine avec le compilateur `lf95`, options de debug et constitution d'une bibliothèque suivant la liste de programmes du fichier `list`.

compile2k_list : Compilation des routines de la liste du fichier `list` de l'environnement, avec le compilateur `lf95`.

evol2k_pms : Evolution d'un modèle incluant la PMS.

evol2k_zams : Evolution d'un modèle initialisé sur la ZAMS homogène.

exe2k : Exécution d'un programme avec le compilateur `lf95`.

exe2k-dbg : Exécution d'un programme avec le compilateur `lf95` et options de debug.

lib_del : Suppression de routines d'une bibliothèque.

lib_del_repl : suppression et remplacement de routines d'une bibliothèque.

lib_repl : Remplacement de routines d'une bibliothèque.

on_voit : Création d'un fichier postscript.

rempl2k_mod : Remplacement d'un module et formation de l'exécutable pour exploitation.

rempl2k_mod-dbg : Remplacement d'un module et formation de l'exécutable pour debug.

pgplot-5.2.0-2.i686.rpm : Source RPM de PGPLOT pour LINUX.

F.5 Sous-directory SUN_STAR_DATA

*****.pms** : Fichiers ASCII d'initialisation de modèles de pré-séquence principale homogène.

Alrd*** : Opacités OPAL "Allard 96".

C95*** : Opacités OPAL "Cox 95".

COX*** : Opacités OPAL "Cox".

EOS*** : Données pour l'équation d'état OPAL.

GN*** : Opacités OPAL "Grevesse Noels".

GS*** : Données pour les opacités OPAL, *cf.* §8.136 (Page 212).

Gz*** : Données pour les opacités OPAL, *cf.* §8.136 (Page 212).

IEOS*** : Données pour l'équation d'état OPAL.

W95*** : Opacités OPAL "Weiss", *cf.* §8.136 (Page 212).

ZFSinterppeos.f : Programme de création de tables d'équation d'état OPAL.

Z_interp_IEOS : Programme de création de tables d'équation d'état OPAL.

ascii2bin_opa : Programme de création de tables d'opacité OPAL en binaire.

datai_gr2.tar.gz : Données pour les accélérations radiatives Alécian2, *cf.* §8.58 (Page 170).

extract_opa_yveline.explik : Explications pour la création de fichiers d'opacité exploitables par `opa_yveline.f`.

fesh* ** : Données pour les tables de loi $T(\tau)$ de type `roger**`, *cf.* § 8.118 (Page 201).

m*.zams.gz** : Fichiers ASCII d'initialisation de modèles de séquence principale d'âge zéro homogène.

mhd1.tab* ** : Données pour l'équation d'état MHD, *cf.* § 8.53 (Page 166).

opa_yveline.data.gz : Tables d'opacité Yveline. *cf.* § 8.83 (Page 185)

opal* ** : Fichiers ASCII pour la création de tables d'équation d'étatOPAL.

opint_v9f.tar.gz : Package d'opacité Houdek, *cf.* § 8.79 (Page 183),

peos* ** : Tables pour l'équation d'état OPAL.

phi_psi2* ** : Données pour le calcul des accélérations radiatives Alécian1, *cf.* § 8.5 (Page 144).

Bibliographie

[1998] Adelberger E., et al., 1998, Rev. Mod. Phys. 70, 4, 1265

[1989] Anders E., Grevesse N., 1989, Geochimica et Cosmochimica Acta 53, 197

[1999] Angulo C., Arnould M., Rayet M., and the NACRE collaboration 1999, Nuclear Physics A 656, 1 and WEb site http ://pntpm.ulb.ac.be/Nacre/nacre.htm

[1985] Alécian G., 1985, A&A 145, 275

[1990] Alécian G., Artru M.C., 1990, A&A 234, 323

[1990] Alécian G., Michaud G., Tully J., 1990, ApJ 411, 882

[1969] Arnett W.D., Truran J.W., 1969, ApJ 157, 339

[1969] Burgers J.M., 1969, Flow equations for composite gases, Academic Press, New york and London

[1988] Caughlan G.R., Fowler W.A., 1988, Atomic Data and Nuclear Data Tables 40, 284

[1991] Canuto V.M., Mazitelli I., 1991, ApJ 370, 295

[1996] Canuto V.M., Goldman ? ? ?, Mazitelli I., 1996, ApJ 473, 550 **INCOMPLET**

[1988] Christensen-Dalsgaard J., 1988, Computational procedures for GONG solar model project, Astronomisk Institut, Aarhus Universitet

[1968] Clayton D.D. ,1968, Principles of Stellar Evolution and Nucleosynthesis, Mc Graw–Hill

[1987] Conte S.D., de Boor C., 1987, Elementary Numerical Analysis, McGraw-Hill Book Company, third edition

[1968] Cox J.P., Guili R.T., 1968, Principles of Stellar Structure, vol I & II, Gordon & Bretch, New-York

[1989] Cox A.N., Guzik J.A., Kidman R.B., 1989, ApJ 342, 1187

[1978] De Boor C., 1978, A Practical Guide to Splines Springer ; third ed. 1985

[1971] Eggleton P., 1971, Mon. Not. Roy. astr. Soc. 151, 351

[1972] Eggleton P., 1972, Mon. Not. Roy. astr. Soc. 156, 361

[1973] Eggleton P.P., Faulkner J., Flannery B.P., 1973, A&A 23, 325

[1975] Fowler W.A., Caughlan G.R., Zimmerman B.A., 1975, Ann. Rev. Astron. Astrophys. 13, 69

[1997] Gabriel M., 1997, A&A 327, 771

[1997] Grevesse N., Noels A.,**INCOMPLET À VÉRIFIER**

[1991] Hairer E., Wanner G., 1991, Solving Ordinary Differential Equations II, Springer-Verlag

[2002] Heiter ? ? ? et al ? ? ? ?, 2002, A&A **INCOMPLET**

[1962] Henrici P., 1962, Discrete Variable Methods in Ordinary Differential Equations, John Wiley & Sons.

[1959] Henyey L.G., Le Levier R., Levee R.D., 1959, ApJ 129, 2

[1965] Henyey L.G., Vardya M.S., Bodenheimer P., 1965, ApJ 142, 841

[1996] Houdek G., Rogl J., 1996, Bull. Astr. Soc. India 24, 317

[1964] Iben I., 1964, ApJ 141, 993

[1975] Iben I., 1975, ApJ 196, 525

[1985] Iben I., MacDonald J., 1985, ApJ 296, 540

[1991] Iglesias C.A., Rogers F.J., 1991, ApJ 371 ,408

[1979] Kaps P., Rentrop P., 1979, Numer. Math. 33, 55

[1968] Kippenhahn R., Weigert A., Hofmeister E., 1968, Methods of Computational Physics 7, 129

[1991] Kippenhahn R., Weigert A., 1991, Stellar Structure and Evolution, Springer-Verlag

[1968] Lang Astrophysical formulae A COMPLETER aussi dans la bibli de CESAM2k

[1983] Maeder A., 1983, A&A 120, 113

[1987] Maeder A., Meynet G., 1987, A&A 182, 243

[1998] Maeder A., Zahn J.P., 1998, A&A 334, 1000

[1985] Marchouk G., Agochkov, V., 1985, Méthode des Eléments Finis, Mir

[1978] Mihalas D., 1978, Stellar Atmosphere, 2d Ed. Freeman and Cie

[2004] Mathis S., Zahn J.P., 2004, A&A

[2004] Mathis S., Palacios A., Zahn J.P., 2004

[1988] Mihalas D., Däppen W., Hummer D.G., 1988, ApJ 331, 815

[1993] Michaud G., Proffitt C.R., 1993, *Inside the Stars*, ed. A. Baglin & W.W. Weiss (San Francisco : ASP), 246

[1977] Mitler H.E., 1977, ApJ 212, 513

[1994] Morel P., van't Veer C., Provost J., Berthomieu G., Castelli F., Cayrel R., Lebreton Y., 1994, A&A 286, 91

[1997] Morel P., 1997, A&AS 124, 597 (une version incluant les appendices, est disponible sur le WEB : http ://www.obs-nice.fr/morel/articles.html)

[1999] Morel P., Pichon B., Provost J., Berthomieu G., 1999, A&A 350, 275

[2002] Morel P., Thévenin F., 2002, A&A 390, 611

[1989] Muchmore D., 1984, ApJ 278, 769

[2003] Palacios A., Talon S.,Charbonnel C., Forestini M., A&A 2003

[1986] Paquette C., Pelletier C., Fontaine G., Michaud G., 1986, ApJS 61, 177

[1986] Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., 1986, Numerical Recipes, Cambridge University Press, Cambridge

[1991] Proffit C.R., Michaud G., 1991, ApJ 371, 584

[1994] Quarteroni A., Valli A., 1994, Numerical approximation of partial differential equations, Springer-Verlag

[1967] Richtmyer R.D., Morton K.W., 1967, Finite Differences Methods for Initial Values Problems, 2d. Ed., Intersciences Publishers, New-York

[1981] Schatzman E., Maeder A., Angrand F., Glowinski R., 1981, A&A 96, 1

[1990] Schatzman E., Praderie F., 1990, Les Etoiles, Ed. InterEditions/Editions du CNRS

[1981] Schumaker L., 1981, Splines Functions : Basic Theory, John Whiley

[1979] Stoer J., Bulirsch R., 1979, Introduction to Numerical Analysis, Springer-Verlag

[1970] Strittmatter P.A., Faulkner J., Robertson J.W., Faulkner D.J., 1970, ApJ 161, 369

[1997] Talon S., Zahn J.P., Maeder A., Meynet G., A&A 322, 209

[1994] Thoul A.A., Bahcall J.N., Loeb A., 1994, ApJ 421, 828

[1980] Trenoguine V., 1980, Analyse Fonctionnelle, Mir

[1969] Wagoner R.V., 1969, ApJS 18, 247

[1980] Winkler K.H., Newman M.J., 1980, ApJ 236, 201

[1991] Zahn J.P., 1991, A&A 252, 179