Stellar evolution code CESAM2*k*

Quick reference: Installation, Usage

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

Version V3.2, 20 May 2008.

Qund 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

This document was translated from French into English by Daniel Reese, whose work is supported by the European Helio- and Asteroseismology Network (HELAS), a major international collaboration funded by the European Commission's Sixth Framework Programme. D. R. wishes to thank Kara Burke for proofreading this document and correcting a number of mistakes.

Contents

List of Figures

List of Tables

Index

Chapter 1

Installation

On n'exécute pas tout ce qui se propose
Et le chemin est long du projet à la chose.

Le Tartuffe.

1.1 CESAM2*k*

The way in which CESAM2*k* is used is different from that of previous versions of CESAM. The executable, `cesam2k.out`, is created only once. It is in the data file, `mon_model.don`, that the user specifies which physics he wants to implement as well as the main parameters which define the model to be calculated. As a result, it is no longer necessary to create a new executable for each usage, as was the case in previous versions of CESAM. It is nonetheless possible to modify most options which are defined in the source code, thanks to "adjustment" files which are described in § 3 (Page 39), and which enable the user to adjust CESAM to different computational capacities as well as to most characteristics of the computed models.

CESAM2*k* is programmed in `FORTRAN95`, which enables the use of dynamic arrays adapted to particular needs. The program has successfully been compiled using:

- LINUX: INTEL's `ifc`, Lahey-Fujitsu `lf95`,
- UNIX: UNIX compilers for ALPHA processors
- WINDOWS: Compact Visual Fortran

The version number `CESAM2k_Vi.j.k` is defined in the following way:

i increases by one for each modification of the binary resumption files *.pms, *.rep, *.dat *etc...* . Indexes **j** and **k** are then set to 0.

j increases by one for each modification of one of the ASCII data files *.don or customisation files. Index **k** is then set to 0.

k increases by one for each noteworthy modification.

The file "`journal`", located in the subdirectory `SOURCE`, contains a brief chronological description of the different modifications.

1.1.1 PGPLOT

CESAM2k uses the graphics software package PGPLOT, which has not been configured to work with the compiler `ifc`. Thierry Corbard, Observatoire de la Côte d'Azur, has written the necessary patch, which can be found in the subdirectory `SCRIPTS`. It is nonetheless possible to avoid using this software, *cf.* § A.2.3 (Page 61)

1.2 Distribution contents

The software package `CESAM2k`[1] contains the following subdirectories:

- `EXPLOIT`: a set of programs and example files used for running the program.
- `NOTICE`: LaTeX and postscript files which contain the reference manual and the quick reference.
- `SCRIPTS`: set of `UNIX/LINUX` procedures, including a `makefile`, for installing and running the program in the `csh` shell.
- `SOURCE`: the set of subroutine, module and program source files.
- `SUN_STAR_DATA`: a set of physical data, tabulated EOS for example.
- `TESTS`: test programs[2].

1.3 Installation

We now describe the installation under `UNIX/LINUX` using a C-shell.

PB `When using a compiler which does not fully comply with the F95 norm (like some of INTEL's compilers`[3]`), it is not possible to allocate memory in a subroutine outside the place where the allocatable array is declared. In such a situation, it is necessary to replace the instruction ALLOCATABLE by POINTER when declaring these arrays` *cf.* § A.1 (Page 59).

- Extract the distribution:

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

  The subdirectory `CESAM2k` is created[4]. It contains 6 subdirectories which are tarred and compressed:

[1] 2k stands for "2000".
[2] Some of these programs still need to be adapted to the current version of CESAM2k.
[3] Contrary to what is generally thought, this compiler is not free, *cf.* the agreement licence.
[4] Version numbers and dates are voluntarily left out of this reference manual.

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

It is best to extract these only as needed. This then enables one to put the subdirectories in the most appropriate locations and hence to facilitate making updates, which usually affect only the subdirectory `SOURCE`.

- In the subdirectory `SCRIPT`, adjust the parameters in the compilation and execution procedures, `compile2k` and `exe2k` respectively, so as to work with the F95 compiler; also enter the paths to the source and the libraries. Make the procedures executable: `chmod 700 *`.

- When working with the `csh` or `tcsh` shell, enter the paths to the library and executables in the `.cshrc` file. For example: `set mypath=(.... ~/CESAM2k/SOURCE ~/CESAM2k/EXPLOIT .....)`

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

  When working with the `bash` shell, enter the following instructions in the `.bashrc` file:

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

- If it has not already been done, install the graphics software `PGPLOT`[5], and enter its library path in the `.cshrc` file:

  `LD_LIBRARY_PATH **** :/usr/lib/pgplot`

  and define the environment variable [6]:

  `setenv PGPLOT_DIR /usr/lib/pgplot`.

  § A.2.1 (Page 60) gives more information on installing this software.

  In case you are unable to install this software, carry out the following corrections in the subdirectory `SOURCE`:

  - Remove `boite`, `box`, `des_r`, `des_m` from the `INCLUDE` list in the `mod_numerique` module.
  - In the `des` subroutine, only keep `CASE ('no_des')` and possibly `CASE DEFAULT`.
  - In the `sortie` subroutine, remove `CALL pgend`.

PB `This list of corrections may be incomplete.`

[5] `PGPLOT`, along with its reference manual, can be downloaded via `anonymous ftp` from `http://astro.caltech.edu/~tpj/pgplot/`.

[6] One of these instructions may be redundant. The opinion of experts is welcome.

1.4 Adapting the source code

Before creating the executable module, it is necessary to adjust certain free parameters in the source code so that it will run on the computer you are using.

- On line 105 of subroutine `des_m`, adjust the name of the "device": `device='/xw'`, or put `device='?'` in order to obtain the list of available devices.
- Do the same thing on line 89 of subroutine `des_r`.

In the subroutines `des_m` and `des_r`, it may be necessary to adjust the height and width of the frames, in order to centre them better in the window, or to customise the plot *cf.* § 3.10 (Page 50).

1.5 Creating the executable module

Go to subdirectory `SOURCE`, and run the procedure[7] `genere_cesam2k`.
After compilation, one obtains the list of "`object module`" in the `libcesam2k.a` library and the starting menu of 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
 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.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
```

[7] In what follows, the symbol <== means a computer instruction to type in.

```
Pour initialiser un modèle de ZAMS : taper 2 puis RETURN
Pour initialiser un modèle de PMS : taper 3 puis RETURN
 ARRET
```

Depending on the choice of parameters and the type of installation, compilation of the library and the executable module can take from 3 to 10 minutes.
It is also possible, using a similar method, to create a debug executable using the procedure `genere_cesam2k-dbg` in the `SCRIPTS` subdirectory.

1.5.1 Use of a MAKEFILE

It is obviously possible to compile CESAM2k via a `MAKEFILE`. Th. Corbard wrote one located in the `SCRIPT` subdirectory, which is given as an example in § A.3 (Page 62). In order to use it, put `makefile` in the `SOURCE` subdirectory. Adapt its variables to comply with the compiler and the different paths. Finally, type in `make`, and the executable module `cesam2k.out` is automatically created.

1.5.2 List of modules and their functions

List of modules of CESAM2k and their functions:

- Module `mod_kind`: groups together the data types of the different variables.
- Module `mod_numerique`: groups together purely numerical and various useful subroutines.
- Module `mod_donnees`: groups together most quantities which remain fixed during the stellar evolution.
- Module `mod_variables`: groups together most quantities which vary during stellar evolution.
- Module `mod_etat`: groups together the subroutines which deal with the equation of state.
- Module `mod_opa`: groups together the subroutines which deal with calculating the opacity.
- Module `mod_conv`: groups together the subroutines which deal with convection.
- Module `mod_atm`: groups together the subroutines which create the atmosphere.
- Module `mod_nuc`: groups together the subroutines which deal with thermonuclear reactions.
- Module `mod_bp_for_alecian`: groups together the subroutines which deal with radiative accelerations, based on G.Alécian's equations.
- Module `mod_evol`: groups together the subroutines which deal with the time evolution of the chemical composition.

- Module `mod_static`: groups together the subroutines which deal with solving the quasi-static equilibrium.
- Module `mod_cesam`: groups together the subroutines which deal with managing the calculations.
- Module `mod_exploit`: groups together subroutines which deal with processing the results.

Figure 1.1 (Page 17) shows a flow chart of the program. Each `PRIVATE` and/or `PUBLIC` subroutine is introduced through `INCLUDE` statements in its module, the name of which is given in the comments at the head of each subroutine.

1.6 Testing the code

Copy the files `mon_modele.don` and `m010.zams` from the `EXPLOIT` subdirectory into the `TESTS` subdirectory, from which we will work. Edit the file `mon_modele.don`.

- Replace ‘‘`des_m`’’ by ‘‘`no_des`’’ if `PGPLOT` is not operational
- adjust ‘‘`nom_chemin`’’ so as to match the path of the `SUN_STAR_DATA` subdirectory of your installation.
- restore the following instructions:

```
&NL_CESAM
NOM_CHEMIN='~/SUN_STAR_DATA/',    <==== to be adjusted
NOM_CTES='ctes_94',
NOM_DES='des_m',   <==== to be adjusted
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
W_ROT=0.d0,
UNIT='jours',
NOM_DIFFW='diffw_0',
NOM_THW='rot_0',
NOM_PERTW='pertw_0',
P_PERTW=0.d0,
LIM_JPZ=.TRUE.,
NOM_DES_ROT='no_des'
/
&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.
 /
```

Run `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:
```

The test has now finished.

When a figure is created directly on line, depending on the screen's resolution, frames are sometimes truncated and/or do not make satisfactory use of the available space. In such a situation, one needs to adjust the dimensions of the frames in the subroutines `des_m` and `des_r` from the `SOURCE` subdirectory, *cf.* § 1.4 (Page 4). Optimal values can be found by using the file `device` which allows one to customise the figure, *cf.* § 3.10 (Page 50).

1.7 Adjusting the format of physical data

CESAM2k uses physical data *i.e.* EOS, opacities *etc...* stored in binary files that need to be generated from `ASCII` data files in the `SUN_STAR_DATA` subdirectory. In what follows, we will work from this subdirectory.

Binary data files created with previous versions of CESAM ***can be used by*** CESAM2**k**.

PB `It is impossible to avoid carrying out calculations using opacity tables and an equation of state in which the heavy elements' abundances` Z `do not correspond to the values used in the model.`

1.7.1 "Yveline" opacities

Decompress the files `opa_yveline.data.gz` and `ascii2bin_opa.f.gz`: `gunzip opa_yveline.data.gz ascii2bin_opa.f.gz`

Do the ASCII ==> binary transformation by running[8] the program `ascii2bin_opa`: `exe2k ascii2bin_opa`

At the question:

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

one obtains:

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

answer:

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

one obtains:

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

answer:

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

on the next question answer "c". One can also recompress the ASCII file and erase the executable. The following is obtained:

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

Recompress the ASCII data: `gzip opa_yveline.data <==`
and erase the executable: `rm ascii2bin_opa.out <==`
One can find supplementary information in the file `extract_opa_yveline_explik`.

[8] With F95, it is useful to use the procedures from the `SCRIPTS` subdirectory.

1.7.2 Opacities for $Z > 0.1$

Type 1 OPAL opacities only exist for metal abundances below $Z < 0.1$. The `z14xcotrin21` package by Arnold I. Boothroyd is able to take into account the nucleosynthesis of heavy elements by taking into account type 2 OPAL opacities. Even though it hasn't been implemented for all the different cases, ***its USE remains DELICATE***. During the first run, the system might ask you to decompress the subroutines from the `SUN_STAR_DATA` subdirectory.

PB `The use of this subroutine is only necessary when the local temperature is below 7Kev. Below this limit, the medium is assumed to be completely ionised and the Rosseland opacity reduces to Compton diffusion (Cox & Giuli, 1968, par. 16.6).`

1.7.3 HOUDEK opacities

OPAL opacity calculation package, (Iglesias & Rogers, 1991), based on the birational splines interpolation package from Houdek & Rogl (1996); it is used by the external subroutine `opa_houdek9`. This package (`opint_v9f.tar.gz`) can by obtained via anonymous ftp from `ftp.ast.cam.ac.uk` and is located in the subdirectory `pub/hg/`. This package needs to be decompressed and extracted from the `opint_v9f.tar.gz` file, also located in the `SUN_STAR_DATA` subdirectory:

```
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
```

It is then safer to recompress the file `opint_v9f.tar`. Using the instructions in README, edit the makefile in the `v9` directory and adapt the compiler and optimiser so as to work with the UNIX computer you are using.

Run the MAKEFILE : `make`.
In the file `v9/OPINPATH_AX`, replace the "./" by the paths `./v9/opal95/opal95e.bin`, for example:

`/home/bilou/SUN_STAR_DATA/v9/opal95/opal95e.bin`

do the same for `v9/OPINPATH_91` and `v9/OPINPATH_95`.

PB `In the calling subroutine opa_houdek9, the PATH to the data files: tabnam=TRIM(path)//TRIM(f_opa(2)) makes use of the` **second** `opacity file f_opa(2), as is coded in the NAMELIST NL_OPA from the text described in` § 2.3 (Page 20).
Define the parameters `iorder` and `imode` by referring to the reference file `./doc/notes.ps` in the `v9` directory. In the script file `exe2k`, set LINK to the library's path `libopint.a`.

PB `The computer implementation of opa_houdek9 is` **delicate**.

1.7.3.1 Simplified implementation by B. Pichon

B. Pichon from the OCA has simplified, and especially made more robust, the installation of Houdek's original package.

Go into the `HOUDEK` sub-directory, and successively run the following procedures:

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

Modify and adapt the files `OPINPATH` as described above. Data is available on request from `Bernard.Pichon@oca.eu`.

1.7.4 Equation of state OPAL

1.7.4.1 OPAL 1991

The equation of state OPAL (Iglesias & Rogers, 1991) is called by the subroutine `etat_opal`. The source and tables of the equation of state were taken from the website `ftp://www-phys.llnl.gov/pub/opal/eos/`, where one can also find a README which explains how to construct an ASCII table for the equation of state at a given metallicity Z.

PB `Some of the subroutines from the original OPAL package have been modified so as to enable their use with` CESAM2*k*; `in particular, the blockdata has been removed.`

The compressed ASCII files `peos*.gz` contain data for the equation of state OPAL. One needs to go into the subdirectory `SUN_STAR_DATA` and decompress these files: `gunzip peos*`.

Run the program `ZFSinterppeos` after having decompressed the source file: `ZFSinterppeos.f.gz`

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

enter a value for Z, the mass fraction of metal abundance; the result is:

```
          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
```

The ASCII file `EOSdata` takes up approximately 9Mb when it is created.
Edit the source code to the program `opal_ascii_bin`, and choose a name for the binary file, for example:

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

     CLOSE(unit=60)   <===   line 117 of 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)

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

Run the modified program `opal_ascii_bin`:

```
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
```

Remove the executables:

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

Finally, recompress the ASCII files, and remove the files `EOSdata` and `fort.2` which are no longer useful:

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

1.7.4.2 OPAL 2001

The equations of state 2001 offer two choices for metallicity: either one creates a table for a given metallicity like in the 1991 version, and the hydrogen abundance must fall in the range $X \in [0, 0.8]$, or $X \in [0, 1]$ but $Z = 0$, *cf.* § 2.7 (Page 26).

- For $Z \neq 0$, go into the `SUN_STAR_DATA` subdirectory, and create the file for the equation of state corresponding to the required metallicity by running the program `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  <===== enter a value for Z
     0.025000000
  bilou@port-morel%
  ```

  A compressed ASCII file (`EOSdata.gz`) is then created. Transform this file into binary format by giving it the name which will identify it in what follows. In order to do this, enter this name in the program `opalZ_ascii_bin` (towards line 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'
```

then run this program, in which there is a reading test:

```
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%
```

One can then remove `EOSdata.gz`:

```
rm EOSdata.gz
```

- For $Z = 0$, the interpolation uses the file `EOSdata_H-He` which needs to be transformed into binary format by running the program `opalX_ascii_bin` from the `SUN_STAR_DATA` subdirectory.

1.7.5 MHD equation of state

MHD equation of state package (Mihalas et al., 1988) called by the subroutine `etat_mhd`. This package was supplied by W. Däppen. The 8 tables `mhd1_oc_tau.tab` ... `mhd8_oc_tau.tab` which can be found on the anonymous server usc.edu, in the directory `pub/astro-physics/mhd-oc-tau/wd-evo`, are also located in the `SUN_STAR_DATA` subdirectory under the names `mhd1.tab.gz` ... `mhd8.tab.gz`. One needs to go into the subdirectory `SUN_STAR_DATA` and decompress these ASCII tables: `gzip mhd* <==` and the source code to the program `fmttob`: `gzip fmttob.f.gz`. Transform these tables into binary format by running `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%
```

Remove the executables:

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

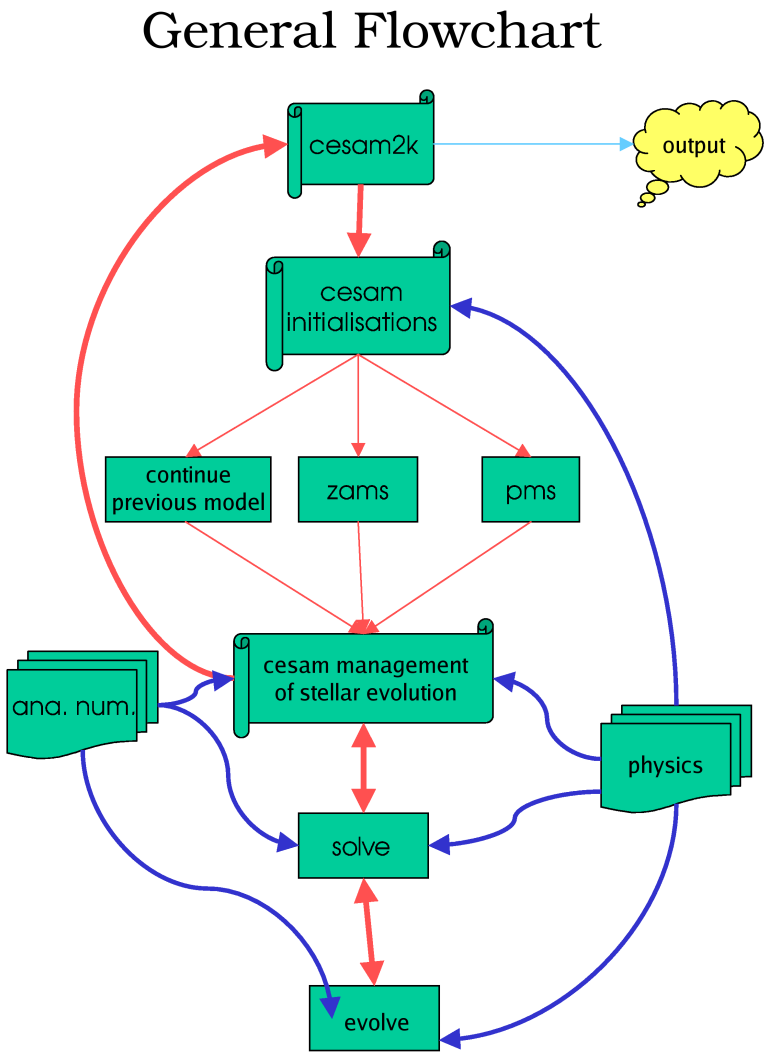

Figure 1.1: Main flow chart for CESAM2k. The program `cesam2k` is compiled only once. It calls the subroutine `cesam` which is, in fact, the main program. After doing some initialisations, `cesam` follows one of 3 possible courses: pursue a star's evolution, initialise a homogeneous ZAMS star, initialise a PMS star. The program then goes on to do various operations: management of the stellar evolution, listing, creation of output files, plot, etc... Afterwards, `resout` is called, in which the program alternates between solving the quasi-static equilibrium equations and solving the angular momentum and chemical composition evolution equations. The latter two are solved separately in `evol`. Once the results have converged, the program goes back to the management operations of `cesam`, and from there either pursues stellar evolution or returns to `cesam` in order to exit. The numerical analysis subroutines and the ones which manage the physics are called at different levels in the calculations.

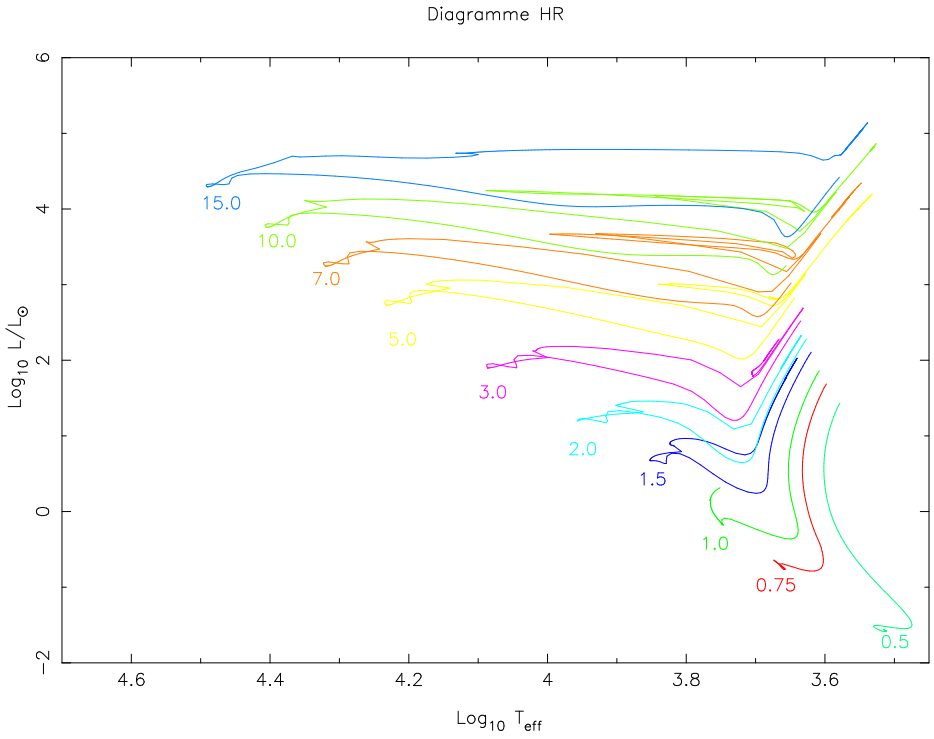

Figure 1.2: Evolutionary tracks which include the pre-main sequence for stars with a mass ranging from $0.5M_\odot$ to $15M_\odot$. These go beyond the helium flash for stars with a mass above $3M_\odot$. For the $3M_\odot$ and $5M_\odot$ models, the calculations were stopped after carbon was used up in the stellar centre. The oxygen destruction phase was reached by the $7M_\odot$ and $10M_\odot$ models. After having exhausted the oxygen in its core, the $15M_\odot$ model reached the limit of validity of the quasi-equilibrium hypothesis. These tracks were obtained using the data file described in § 2.3 (Page 20).

.

Chapter 2

Usage

Va despacio, estoy de prisa.

Mexican Proverb

2.1 Advice

No matter how well it is written, a numerical code can never be completely "ready-to-use". Before starting to use the code, one needs to bear in mind the following remark:

> *A program that doesn't function, or functions poorly, in simple cases will not function, or will function worse, in complicated cases.*

A code on internal structure does not, alas, escape from this rule. In this respect, even though several types of standard "precision" are implemented in CESAM2k, it is almost always necessary to optimise the numerical parameters in function of the conditions which occur during the calculations thanks to various "adjustments". To tackle a problem by introducing *ad initio* all the complexity which one wishes to reach is a waste of time. In order to acquire experience on how the evolution proceeds, it very quickly becomes necessary to work with simple cases. It is recommended to simplify the problem by choosing numerical and physical options which are robust and effective, in order to get a feel for what is happening, and then to gradually introduce the desired complexity.

PB `Below, we only give a brief description of various subroutines. For more information, please read the comments in the subroutines.`

2.2 The data file: `my_model.don`

The options concerning which physics to implement and the data are sent to the main program by `NAMELISTS` which are grouped together in an ASCII file, the name of which **necessarily** has the extension `.don`, for example: `my_model.don`.

An example file, `mon_modele.don`[1] is given in the `EXPLOIT` subdirectory.

- `& NL_CESAM`: groups together overall calculation conditions.
- `& NL_MASS`: groups together physical parameters to do with the spatial variable.
- `& NL_EVOL`: groups together physical parameters to do with the temporal variable.
- `& NL_CHIM`: groups together physical parameters to do with chemical composition.
- `& NL_CONV`: groups together physical parameters to do with convection.
- `& NL_DIFF`: groups together physical parameters to do with diffusion.
- `& NL_ROT`: groups together physical parameters to do with rotation.
- `& NL_ETAT`: groups together the names of *external* files which may be needed when calculating the equation of state.
- `& NL_OPA`: groups together the names of *external* files which may be needed when calculating the opacity.
- `& NL_NUC`: defines options which relate to calculating thermonuclear reaction rates.
- `& NL_ATM`: groups together physical parameters to do with reproducing the atmosphere.

Usage without diffusion or rotation:
For a classical evolution, *i.e.* without diffusion of chemical elements or rotation, one can remove the `NL_DIFF` and `NL_ROT NAMELISTS`, in order to reduce the data file.

2.3 Example of a data file

The following example is deliberately different from the one in § 1.6 (Page 6).

```
&NL_CESAM
NOM_CHEMIN='~/SUN_STAR_DATA/',
NOM_CTES='ctes_94',
NOM_DES='des_m',
NOM_OUTPUT='no_output',
N_MAX=6000,
PRECISION='av'
/
&NL_MASS
```

[1] In French, "`mon_modele`" means "`my_model`.

```
MTOT=2.d0,
NOM_PERTM='pertm_ext',
MDOT=0.d0
/
&NL_EVOL
AGEMAX=12.d3,
ARRET='else',
DTLIST=1.d10,
LOG_TEFF=10.d0,
NB_MAX_MODELES=700,
HE_CORE=-0.1d0,
T_STOP=2.0d9,
X_STOP=-0.1d0
/
&NL_CHIM
GRILLE_FIXE=.FALSE.,
NOM_ABON='solaire_gs',
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
W_ROT=0.d0,
UNIT='kms/s',
NOM_DIFFW='diffw_mpz',
NOM_THW='cons_loc_mnt_cin',
NOM_PERTW='pertw_0',
LIM_JPZ=.TRUE.,
```

```
NOM_DES_ROT='end_mod'
/
&NL_ETAT
NOM_ETAT='etat_eff',
F_EOS='eos_opal_250.bin',7*' '
/
&NL_OPA
NOM_OPA='opa_houdek9',
F_OPA='opa_yveline.bin','v9/OPINTPATH_AX',6*' '
/
&NL_NUC
NOM_NUC='ppcno3aco',
NOM_NUC_CPL='NACRE',
MITLER=F
/
&NL_ATM
NOM_ATM='lim_atm',
NOM_TDETAU='hopf',
TAU_MAX=30.d0,
LIM_RO=.TRUE.
/
```

PB The names of the subroutines are ***case sensitive***, for example, one should write ‘‘NACRE’’ and not ‘‘nacre’’.

2.4 Arguments of the NL_CESAM NAMELIST

- NOM_CHEMIN='∼/SUN_STAR_DATA/': path and name of the directory which contains the physical data, such as the opacity tables.
- NOM_CTES: gives the name of the subroutine which initialises the main physical constants to be used:
  - ctes_85: physical constants proposed by GONG,
  - ctes_94: more recent physical constants.
  - ctes_94m: same as ctes_94 except that the mass of atomic nuclei have been rounded off to the nearest integer.
- NOM_DES: name of the plotting subroutine to be used:
  - des_m: plot as a function of mass, it uses PGPLOT,
  - des_r: plot as a function of radius, it uses PGPLOT,
  - no_des: no *on-line* plot, used if PGPLOT is unavailable, see § 1.3 (Page 2),
  - zoom: customised plot as described in § 3.9 (Page 48), this can be useful for obtaining a particular condition, such as a specific location in the HR diagram.

- NOM_OUTPUT: type of ASCII file to be produced upon exiting the calculations:
  - osc_adia: generates the ASCII file my_model-ad.osc used for calculating adiabatic pulsations and making some of the plots
  - all_adia: generates ASCII files used for calculating adiabatic pulsations and making plots for ***EACH*** model. These files take on names of the form *nnnn*-my_model-ad.osc, *nnnn* being the number of the model,
  - osc_invers: generates the ASCII file my_model-inv.osc used for inversions,
  - all_invers: generates ***ALL*** of the ASCII files *nnnn*-my_model-inv.osc used for inversions,
  - osc_nadia: generates the ASCII file my_model-nad.osc used for calculating non-adiabatic oscillations,
  - all_nadia: generates ***ALL*** of the ASCII files *nnnn*-my_model-nad.osc used for calculating non-adiabatic calculations, *nnnn* being the number of the model,
  - no_output: does not generate an ASCII file,
  - ascii: customised version of the ASCII file my_model-ascii, which is described in § 3.7 (Page 44),
  - all_ascii: generates ***ALL*** of the customised ASCII files *nnnn*-my_model-ascii, *nnnn* being the number of the model.
- N_MAX=1000: the maximum number of layers[2] is given by MAX(N_MAX,nnnn), nnnn being the maximum number of layers as determined by the routine cesam according to the required precision; similarly, a minimal number of layers is set at 300. If one chooses to output an ASCII file, then picking a negative value for N_MAX allows one to impose |N_MAX| layers on the last model of the evolution sequence.
- PRECISION='av': level of precision required. Table 2.1 (Page 25) gives the values of the numerical parameters used for the different options defined in the program cesam. Suggested values are:
  - 'np' (normal precision) useful for getting a broad idea of the stellar evolution,
  - 'pr' (realistic precision) useful for doing stellar evolution without requiring high precision,
  - 'sp' (super precision) and 'sa' (solar precision) if one requires extreme precision. With 'sa', if one chooses to output an ASCII file, the last models from an evolution sequence are calculated with the maximum number of layers N_MAX,
  - 'av' (advanced stages) useful for doing stellar evolution in the advanced stages without requiring high precision,

[2]The nature (REAL, INTEGER, CHARACTER, ...) of the different numerical and logical parameters is deduced from values which are given as an example.

– `'mx'` (maximum number of layers) identical to `'sp'` in which the number of layers is fixed, and is equal to `N_MAX`;

The "mesh factors"[3] are set according to the type of precision. The most common values are: `ctel=0, ctep=-1, ctem=15, cter=0, ctet=-1`. To use different parameters, see § 3.8 (Page 45).

2.5 Arguments of the NL_MASS NAMELIST

- `MTOT=1.d0`: total initial mass, in $M_\odot$,
- `NOM_PERTM`: name of the subroutine which implements mass loss:
  - `pertm_ext`: linear external mass loss/gain as a function of time,
  - `pertm_msol`: linear external mass loss/gain as a function of time, the stellar mass remains above or equal to $1\,M_\odot$,
  - `pertm_tot`: linear external mass loss/gain as a function of time; takes into account mass loss due to thermonuclear reactions ($E = mc^2$),
  - `pertm_waldron`: empirical mass loss from Waldrom (1985).
- `MDOT=1.d-14`: rate of mass loss/gain, in $M_\odot$/year. Positive values correspond to mass gain and negative values to mass loss. In the standard case, the chemical composition of the mass loss/gain is that of the model's outermost layer. If need be, § 3.5 (Page 42) explains how to customise this chemical composition.

2.6 Arguments of the NL_EVOL NAMELIST

- `AGEMAX=4.5d4`: upper age limit in millions of years,
- `ARRET='post'`: stop at the ZAMS (`'zams'`), at the end of the ZAMS (`'post'`), when helium combustion starts (`'cohe'`) when carbon combustion starts (`'coca'`) when oxygen combustion starts (`'coox'`) somewhere else (`'else'`).
- `DTLIST=1.d3`: minimal time span, in millions of years, between consecutive detailed lists describing the model (`my_model.lis` file),
- `LOG_TEFF=-3.7d0`: stop if that value for $\log_{10} T_{\rm eff}$ is exceeded when `log_teff` is positive, or if opposite occurs when `log_teff` is negative,
- `NB_MAX_MODELES=500`: stop after having calculated that number of models, even if it means 0 models; if `NB_MAX_MODELES < 0`, ***all*** binary files will be written in the environment, their names being of the form `my_model`***nnnn***`_B.***` where ***nnnn*** is the number of the model and `.***` an extension such as `.hom`,

[3] These are coefficients from a "mesh function" which determines how mesh points are distributed when solving the stellar equations (Eggleton, 1971).

Table 2.1: The values of various numerical parameters for the different types of precision. The values of these parameters can be adapted to the type of evolution by using the option 'rg', *cf.* § 3.8 (Page 45); in this case the parameters given in the file reglages located in the environment are used. The following is a list of different designations and their meanings: pr: for realistic precision, sp: for super precision, lm: for low mass, av: for models which could reach advanced stages, np: for normal precision models, co: for models to be used in interpreting results from the CoRoT satellite. The meaning of the different numerical parameters is explained in § 3.8 (Page 45) and in lines 741-766 of the program cesam in the SOURCE subdirectory. The default parameters are those used for realistic precision pr. Other options for the precision, some of which use Eulerian coordinates, exist but are not explained here.

	pr	sp	lm	av	np	co	sa
m_qs	2				1		
m_ch	2						
m_rot	3				2		
m_tds	2						
m_ptm	2						
ordre	2				1		
precix	10^{-3}	10^{-4}			0.005	10^{-4}	10^{-5}
precit	0.15	0.05	0.2		0.3	0.05	0.02
ro_test	0.1						
psi0	0.08	0.06			0.1	0.06	0.06
d_grav	0.5			50	1.0		
loc_zc	10^{-3}	10^{-4}			0.005	10^{-4}	10^{-5}
dtmax	200	50	300		300	50	50
ini0	4	5	5		3	5	5
n_atm	75	100			50	100	100
kipp	.FALSE.		.TRUE.	.TRUE.	.TRUE.		
en_masse	.TRUE.						
ctel	0.						
ctep	-1.						
ctem	15.						
cter	0.						
ctet=	-1.						
mvt_dis	.TRUE.						
dn_fixe	0.05						
dpsi	0.05						
mu_saha	.TRUE.		.FALSE.	.FALSE.	.FALSE.		
n_max	1000					3000	
ajuste	.TRUE.			.FALSE.	.FALSE.		
lisse	.FALSE.			.TRUE.	.TRUE.		
q0	0.					0.01	0.01
l0	0					5	5
new_bv	.FALSE.						

`.rep`, *etc...* used for the purposes of identification. Initialisation models from the zero age main sequence or the pre-main sequence receive the number 0000; when continuing a previous evolution, the numbering is picked up from where it was left off.

- `HE_CORE=0.1d0`: stop when the helium core has reached a mass of `he_core` ($M_\odot$),
- `T_STOP=1.d7`: stop when this temperature is exceeded in the centre,
- `X_STOP`=0.1d0: stop when this X value is crossed in the centre.

2.6.1 Characteristics of the different types of models

- `'zams'`: A model evolving in the pre-main sequence stage is identified as a ZAMS model as soon as the energy released by nuclear reactions exceeds that resulting from gravitational collapse.
- `'post'`: A model evolving along the main sequence is identified as a post-main sequence model as soon as the central hydrogen abundance goes below 0.01.
- `'cohe'`: A model evolving in the post-main sequence stage is identified as a helium combusting model the moment the central temperature exceeds $1\,10^8$K.
- `'coca'`: A model evolving in the helium combustion stage is identified as a carbon combusting model the moment the central temperature exceeds $6\,10^8$K.
- `'coox'`: A model evolving in the carbon combustion stage is identified as an oxygen combusting model the moment the central temperature exceeds $1\,10^9$K.

2.7 Arguments of the `NL_CHIM` NAMELIST

- `GRILLE_FIXE=.TRUE.`: this sets a fixed grid for the chemical composition and the diffusion of angular momentum,
- `NOM_ABON`: name of the initial mixture[4]:
  - `enhan_al`: Allard's α-enhanced abundances,
  - `enhan_cha`: Chaboyer's α-enhanced abundances,
  - `enhan_w`: Weiss' α-enhanced abundances,
  - `meteorites_ag`: meteoritic abundances from Anders & Grevesse (1989),
  - `meteorites_gs`: meteoritic abundances from Grevesse & Sauval (1998),
  - `mixture`: customised implementation of the abundances as described in § 3.2 (Page 40),
  - `solaire_gn`: solar mixture from Grevesse & Noels (1993),

[4] *cf.* OPAL equation of state: WEB site `http://www-phys.llnl.gov/V_Div/OPAL/opal.html`.

  - solaire_gs: solar mixture from Grevesse & Sauval (1998).

  These abundances are initialised in the abon_ini subroutine.

- MODIF_CHIM=.FALSE.: if there are the following types of files in the environment – modif_mix, rap_iso, planet, or vent which enables one to modify respectively the initial composition, *cf.* § 3.4 (Page 42), the isotopic ratios, *cf.* § 3.3 (Page 41), the chemical composition of planetoids, *cf.* § 3.6 (Page 43) or that of the wind, *cf.* § 3.5 (Page 42) – CESAM2k will ask the user to delete these files before starting its calculations. This ***security measure*** is meant to avoid taking into account files which were accidentally left in the environment. One needs to set MODIF_CHIM=.TRUE. to take these files and the options they contain into account .

- GARDE_XISH=.FALSE.: this parameter will be described in the following subsection,

- X0=0.7: initial H abundance per mass unit,

- Y0=0.28: initial He abundance per mass unit,

- ZSX0=0.d0: initial value of Z/X,

  - The initial abundances of a model are determined from X0, Y0 and ZSX0; due to the relationship $1 = X + Y + Z$, only 2 of these 3 quantities should be used. The initial value for Y is ***always*** Y0; it is broken down into the different isotopes which are kept for describing the evolution of helium,
  - if ZSX0 $\leq$ 0, the initial hydrogen abundance is given by X0; $Z = 1 - X - Y$ is then deduced from X0 and Y0,
  - otherwise, *i.e.* ZSX0 $>$ 0, the initial hydrogen abundance is deduced from Y0 and ZSX0, the value of X0 being ignored.

The initial X value used in the calculations is broken down into the different isotopes which are kept for describing the evolution of hydrogen.
When there is diffusion and/or after a first dredge-up, the chemical composition of the outer layers, *i.e.* the one that is observed, is different from the initial chemical composition. It is then necessary to adjust the initial chemical composition in order to obtain the one that is observed at the end of the evolution. Thanks to F. Thévenin's initiative, CESAM2k contains different tools to make this type of adjustment so as to match observations of the metal/H or [Fe/H] ratios:

- Conserve the Z/X ratio of the initial mixture, instead of ZSX0, which is described in § 2.7.1 (Page 28),

- use the original mixture, *cf.* § 3.2 (Page 40),

- modify the isotopic ratios, *cf.* § 3.3 (Page 41),

- modify the abundance ratios of the initial mixture, *cf.* § 3.4 (Page 42).

2.7.1 Alternative: conservation of metal/H or metal/Z

For an initial mixture, which may be modified or left as it is, *cf.* § 3.4 (Page 42), the alternative `GARDE_XISH=.TRUE.` (*respt.* `GARDE_XISH=.FALSE.`) enables one to fix the initial Z (*respt.* X) value so as to conserve the ratio `metal/H` (*respt.* `metal/Z`). The initial Y value is ***ALWAYS*** the value given in the data file, *i.e.* `Y0`.

Therefore, by coding[5] `GARDE_XISH=.TRUE.`, the value which is used for Z/X will be deduced from the abundance ratios metal/H of the mixture and not that of the data file[6], `ZSX0`. The abundance ratios metal/H will be those of the mixture, whereas the ratios metal/Z will be different from those of the mixture.

Conversely, by coding `GARDE_XISH=.FALSE.`, the value `ZSX0` from the data file will be used. The ratios metal/Z will be those from the mixture[7], whereas the ratios metal/H will be different from those of the mixture. For a ***solar calibration***, where one is aiming to keep a fixed Z/X value while conserving the initial metal/Z ratios, one should set `garde_xish=.FALSE.`

PB `The use of GARDE_XISH=.TRUE.` ***IS TRICKY***. `It is recommended to check that the abundances used in the my_model.lis file do in fact correspond to what one wants.`

PB `One needs to be aware of the fact that modifications of the mixture will not be taken into account either in the opacities or in the equation of state.`

2.8 Arguments of the `NL_CONV` NAMELIST

- `NOM_CONV`: name of the subroutine which calculates the temperature gradients in the convection zones.
  - `conv_a0`: MLT convection, with mixing length $l \to 0$ towards the boundaries between radiation and convection zones,
  - `conv_cgm_reza`: convection as described in Canuto & Mazitelli (1991), with $l = \alpha H_{\rm p}$, taking into account Bernkopf's prescription
  - `conv_cm`: convection as described in Canuto & Mazitelli (1991), with $l = \alpha H_{\rm p}$,
  - `conv_cm_reza`: convection as described in Canuto & Mazitelli (1991), with $l = \alpha H_{\rm p}$, taking into account δ,
  - `conv_jmj`: MLT convection with $l = \alpha H_{\rm p}$ (Böhm-Vitense, 1958).
- `alpha=1.7d0`: mixing length parameter,
- `ovshts=0.05d0`: upper overshoot coefficient; a positive value puts an adiabatic gradient in the overshoot zone, whereas a negative value puts a radiative gradient

[5] `GARDE_XISH_MIX`, `GARDE_XISH_MIX_INI`, `GARDE_ZSX` or `GARDE_ZSX_MIX_INI`, for example, would be better names.

[6] Except in the particular case where the values of Y0 and ZSX0 of the data file correspond **exactly** to those of the mixture.

[7] In accordance with the opacity tables.

- ovshti=0.1d0: lower overshoot coefficient; a positive value puts an adiabatic gradient in the overshoot zone, whereas a negative value puts a radiative gradient
- jpz=.FALSE.: the use of JpZ's prescriptions,
- cpturb=0.d0: turbulent pressure coefficient,
- ledoux=.FALSE.: the use of Ledoux's criteria[8].

PB The use of a non-zero turbulent pressure parameter is delicate.

2.9 Arguments of the NL_DIFF NAMELIST

- DIFFUSION=.TRUE.: microscopic diffusion of chemical elements and possibly angular momentum diffusion will be taken into account
- NOM_DIFFM: name of the subroutine which calculates the coefficients of microscopic diffusion:
  - diffm_br: calculates microscopic diffusion coefficients based on Burgers equations and possibly takes into account radiative accelerations,
  - diffm_mp: calculates microscopic diffusion coefficients based on Michaud & Profitt's simplified equations,
  - diffm_0: sets the microscopic diffusion coefficients to zero.
- NOM_DIFFT: name of the subroutine which calculates the coefficients of turbulent diffusion:
  - difft_nu: calculates turbulent diffusion coefficients which include radiative diffusivity
  - difft_gab: calculates turbulent diffusion coefficients based on a simple prescription by M. Gabriel.
  - difft_sun: turbulent diffusion coefficients under the solar convection zone as in M. Gabriel 1997, A&A 327, 771.
- D_TURB=1.d1: isotropic turbulent diffusion coefficient,
- RE_NU=1.d0: radiative diffusivity coefficient,
- NOM_FRAD: name of the subroutine which calculates radiative accelerations,
  - alecian1: calculates radiative accelerations based on G. Alécian's first approach,
  - alecian2: calculates radiative accelerations based on G. Alécian's second approach,
  - no_frad: radiative accelerations are ignored.

[8] To be used with caution.

2.10 Arguments of the NL_ROT NAMELIST

- W_ROT=50.d0: initial rotation rate.
- UNIT: units used for the initial rotation rate. different choices are available so as to easily adapt to observations:
  - jours: initial period in days, corresponds to the rotation rate of the outermost layer,
  - kms/s: velocity of the outermost layer in kilometres/second, in which the radius of the initialisation model has been used to determine the initial rotational velocity. This radius needs to be adjusted by recalculating several times the model of age 0.
  - rad/s: initial rotation rate in radians/second.
- NOM_DIFFW: name of the subroutine which calculates the coefficients for the diffusion of angular momentum:
  - diffw_p03: angular momentum diffusion coefficients from Palacios et al.
  - diffw_mpz: angular momentum diffusion coefficients from Mathis, Palacios & Zahn.
  - diffw_0: sets the angular momentum diffusion coefficients to zero.
  - diffw_cte: constant angular momentum diffusion coefficients : ($D_{\rm eff} = 300,\ D_{\rm h} = 1\,000\,000,\ D_{\rm v} = 250$).
- NOM_THW: Name of the theory used in determining the evolution of the rotation rate:
  - rot_0: zero angular velocity, the model does not take rotation into account.
  - rot_cte: Constant angular velocity, *i.e.* rigid rotation.
  - cons_glob_mnt_cin: evolution with rigid rotation, and global conservation of angular momentum. At the end of each time step, the rotation rate, which is constant throughout the entire model, is readjusted so as to conserve total angular momentum.
  - diff_tz97: Evolution with diffusion of angular momentum, following the theory of Talon & Zahn (1997).
  - diff_mz04: Evolution with diffusion of angular momentum, following the theory of Mathis & Zahn (2004),
  - cons_loc_mnt_cin: evolution with local conservation of angular momentum and rigid rotation in convection zones.
- NOM_PERTW: Name of the subroutine which calculates the loss of angular momentum,
  - pertw_sch: variation of the rotation rate proportional to Ω^3 (Schumanish),

  - pertw_loc: loss of angular momentum proportional to the kinetic energy of the local rotation,
  - pertw_ptm: loss of angular momentum due to mass loss,
  - pertw_0: no loss of angular momentum.

- P_PERTW=-1.d-9: parameter of angular momentum loss to be used respectively with each of the preceding subroutines

- LIM_JPZ=.TRUE.: use of the boundary condition from Mathis & Zahn (2004): the temporal variation of the angular momentum in a convection zone is assumed to be advected to the adjacent radiation zone. LIM_JPZ=.FALSE.: the continuity of the flux of angular momentum is imposed over the radiation/convection zone boundaries.

- NOM_DES_ROT='end_mod': Controls the generation of ASCII files containing the variables on angular momentum diffusion, *cf.* § B.3 (Page 77). If an on-line plot is not required (NOM_DES=no_des), the postscript file with the plots of the different rotation variables is automatically created.

  - no_des: Do not write files.
  - end_evol: Generates a file at the end of the evolution, the name of which is my_model_coeff_rota.dat.
  - all_mod: Generates a file at the end of each time step, the name of which is ***nnnn***-my_model_coeff_rota.dat, where ***nnnn*** is the number of the model. All files thus created are conserved.
  - end_mod: Generates a file at the end of each time step, the name of which is my_model_coeff_rota.dat.
  - all_iter: Generates a file at the end of each iteration when solving the system of equations which govern angular momentum diffusion; these files are useful for making corrections/improvements.

2.11 Arguments of the NL_ETAT NAMELIST

- NOM_ETAT: name of the subroutine which implements the equation of state:

  - etat_ceff: equation of state from Eggleton et al., with Coulomb corrections (*calls* etat_eff *in case of difficulty*).
  - etat_eff: equation of state from Eggleton et al. (*calls* etat_gong2 *in case of difficulty*).
  - etat_gong1: gong1 equation of state, only H and He are taken into account and are assumed to be totally ionised.
  - etat_gong2: gong2 equation of state, only H and He4 are taken into account.

  - `etat_mhd`: equation of state of Mihalas, Hummer & Dappen (*calls* `etat_eff` *in case of difficulty*), makes use of binary tables `mhd1.bin` ... `mhd7.bin` which are constructed in the `SUN_STAR_DATA` subdirectory, see § 1.7.5 (Page 15).
  - `etat_opal`: `opal` 1991 equation of state, (*calls* `etat_eff` *in case of difficulty*), makes use of the binary tables `eos_opal*.bin` which are constructed in the `SUN_STAR_DATA` subdirectory, see § 1.7.4 (Page 12).
  - `etat_opalZ`: `opal` 2001 equation of state, (*calls* `etat_eff` *in case of difficulty*), makes use of the binary tables `eos_opalZ*.bin` which are constructed for a **fixed Z value** in the `SUN_STAR_DATA` subdirectory, *cf.* § 1.7.4 (Page 12). Compared to the 1991 version, the data in the 2001 version has been expanded and improved, and various bugs have been corrected. The data does not enable one to go beyond $X = 0.8$. The data files 2001 and 1991 are incompatible.
  - `etat_opalX`: `opal` 2001 equation of state, different from `etat_opalZ` by the fact that the hydrogen abundance domain has been extended to $X \in [0, 1]$, in order to calculate models *with diffusion* without extended external convection zones. The price to pay is $Z \equiv 0$, which is not very problematic as the equation of state is not very sensitive to Z. `etat_opalX` makes use of **the** binary table `eos_opalX` which is constructed for **the fixed value Z=0** in the `SUN_STAR_DATA` subdirectory, *cf.* § 1.7.4 (Page 12).

- `F_EOS='eos_opal_250.bin',7*' '`: names of the equation of state files (8 at most).

2.12 Arguments of the `NL_OPA` NAMELIST

- `NOM_OPA`: name of the subroutine which calculates the mean Rosseland opacities:

  - `opa_gong`: simplified opacities (improved Kramers).
  - `opa_houdek9`: Houdek opacities, version 9, (OPAL+Alexander), interpolation by rational B-splines; makes use of binary tables constructed in the `SUN_STAR_DATA/V9` subdirectory, see § 1.7.3 (Page 11)
  - `opa_int_zsx`: opacities OPAL93+Kurucz as joined by Yveline, linear interpolations, very useful for tests.
  - `opa_opal2_co, opa_opal2_cno`: OPAL opacities with corrections for $Z > 0.1$, *cf.* § 1.7.2 (Page 11)
  - `opa_opalCO`: OPAL opacities with corrections for $Z > 0.1$ applied only to C and O.
  - `opa_yveline`: OPAL+Alexander opacities as interpolated and joined by Yveline; makes use of binary tables `opa_yveline*.bin` which are constructed in the `SUN_STAR_DATA` subdirectory, see § 1.7.1 (Page 10).

  - opa_yveline_lisse: OPAL+Alexander opacities as joined by Yveline, linear interpolation or smoothing; makes use of binary tables opa_yveline*.bin which are constructed in the SUN_STAR_DATA subdirectory, see § 1.7.1 (Page 10).

- F_OPA='opa_yveline.bin','/HOUDEK/OPINTPATH_AX95',6*' ': name of opacity files (8 at most).

Above the temperature $80\,10^6$K, matter is completely ionised; CESAM2k simplifies the opacity calculations by using Kramers' free-free equations.

2.13 Arguments of the NL_NUC NAMELIST

- NOM_NUC: name of the subroutine which calculates thermonuclear reaction rates and initialises the chemical composition:

  - pp1: simplified calculation of the PP cycle (doesn't allow microscopic diffusion).
  - pp3: PP reactions, 3 elements H, He3, He4 with H2, Li7, Be7 at equilibrium; intended for doing tests in order to make improvements, allows microscopic diffusion.
  - ppcno9: PP+CNO reactions, 9 elements, H2, Li7, Be7 at equilibrium.
  - ppcno9Fe: PP+CNO reactions, 9 elements+Fe, H2, Li7, Be7 at equilibrium.
  - ppcno10: PP+CNO reactions, 10 elements, H2, Be7 at equilibrium.
  - ppcno10Fe: PP+CNO reactions, 10 elements+Fe, H2, Be7 at equilibrium.
  - ppcno10BeBFe: PP+CNO reactions, 10 elements + Li6, Be9, B11, Fe with H2 and Be7 at equilibrium.
  - ppcno11: PP+CNO reactions, 11 elements, Be7 at equilibrium.
  - ppcno12: PP+CNO reactions, 12 elements.
  - ppcno12Be: PP+CNO reactions, 12 elements + Be9.
  - ppcno12Li: PP+CNO reactions, 12 elements + Li6.
  - ppcno12BeBFe: PP+CNO reactions, 12 elements + Li6, Be9, B11, Fe.
  - ppcno3a9: PP+CNO+3α reactions, 9 elements, H2, Li7, Be7 at equilibrium.
  - ppcno3a12Ne: PP+CNO+3α+carbon reactions, 12 elements including Ne22, H2, Li7, Be7 at equilibrium.
  - ppcno3aco: PP+CNO+3α+carbon+oxygen reactions, 17 elements H2, Li7, Be7 at equilibrium.

  The extent of the tabulation interval is fixed according to the zone where the chosen nuclear network is relevant.

- `nom_nuc_cpl`: name of the compilation of nuclear reactions:
  - `Adelb`: Compilation of Adelberger et al. (1998).
  - `Cau-Fow`: Compilation of Caughlan & Fowler (1988).
  - `NACRE`: NACRE compilation (Angulo et al., 1999).
- `MITLER=.FALSE.`: screen effect according to Mitler (1997).

2.14 Arguments of the `NL_ATM` NAMELIST

- `NOM_ATM`: name of the subroutine which produces the atmosphere:
  - `lim_atm`: produces an atmosphere based on a $T(\tau)$ law,
  - `lim_gong1`: produces a simplified atmosphere, GONG1 case,
  - `lim_tau1`: produces a simplified monolayer atmosphere.
- `NOM_TDETAU`: name of the $T(\tau)$ law which is used in producing the atmosphere:
  - `edding`: totally radiative Eddington $T(\tau)$ law.
  - `hopf`: totally radiative Hopf $T(\tau)$ law.
  - `K5750`: non-totally radiative $T(\tau)$ law, derived from the Kurucz solar atmosphere model for $T_{\rm eff} = 5750\,{\rm K}$.
  - `K5777`: non-totally radiative $T(\tau)$ law, derived from the Kurucz solar atmosphere model for $T_{\rm eff} = 5777\,{\rm K}$.
  - `roger00`: non-totally radiative $T(\tau)$ law, derived from Kurucz atmosphere models, with metallicity $[\frac{\rm Fe}{\rm H}] = 0.0$.
  - `roger02`: non-totally radiative $T(\tau)$ law, derived from Kurucz atmosphere models, with metallicity $[\frac{\rm Fe}{\rm H}] = +0.2$.
  - `roger05`: non-totally radiative $T(\tau)$ law, derived from Kurucz atmosphere models, with metallicity $[\frac{\rm Fe}{\rm H}] = -0.5$.
  - `roger10a`: non-totally radiative $T(\tau)$ law, derived from Kurucz atmosphere models, with metallicity $[\frac{\rm Fe}{\rm H}] = -1.0$ with α enhanced elements.
- `TAU_MAX=10.d0`: optical depth at which the atmosphere is connected to the envelope
- `LIM_RO=.TRUE.`: external boundary condition on density, or else on pressure.

2.15 Data files from preceding versions

CESAM2k can process data files from CESAM4 and CESAM5[9]; in these preceding versions, it was necessary to create an executable module according to which physics was implemented. In order to process a data file from CESAM4 or CESAM5 using CESAM2k, it is necessary to indicate which physics to use and to introduce the missing data. CESAM2k does this by using the data and physics options given in the file `physique45` placed in the calculation environment. An example of a `physique45` file is given in § 3.11 (Page 50). When such a file is absent, the data and physics options are given by default in the subroutine `lit_nl_45`, which is subordinated to the subroutine `lit_nl`. These physics options are:

```
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_DIFFW='diffw_0',
NOM_THW='rot_0',
NOM_PERTW='pertw_0',
NOM_DES_ROT='no_des'
NOM_ETAT='etat_eff'
NOM_OPA='opa_yveline'
NOM_NUC='ppcno9'
NOM_NUC_CPL='NACRE'
NOM_ATM='lim_atm'
NOM_TDETAU='hopf'
```

CESAM2k also accepts data files from its first versions used for evaluation.
The purpose of processing old data files is to enable users to pursue previous calculations. As a result, only the functionalities of the older versions can be taken into account.

2.16 Usage under UNIX & LINUX

Under UNIX/LINUX, it is recommended to create the executable module `cesam2k.out` in the `SOURCE` subdirectory and to leave it there. When using CESAM2k, make a separate subdirectory for each application in which to put the data files (`my_model.don`), the initialisation files such as `m020.zams` and ,if need be, adjustment and customisation files.
For example, to calculate models of ξ Hya ($3M_\odot$) from the ZAMS to 400 My:

- Create a subdirectory `KSI_HYA`

[9] This option no longer exists in the more recent versions of CESAM2k.

- Copy the files `m020.zams`[10] and `my_model.don` from the `EXPLOIT` subdirectory.
- Rename the data file `ksi_hya.don`.
- Adjust the physics and parameters of this file to ξ Hya.

Launch two successive calculations of a preliminary ZAMS model `(AGEMAX = 0.d0)` adjusted to the chosen parameters and physics:

1. the first with reduced precision `(PRECISION='np')` using the ASCII initialisation model,
2. and the second with `(PRECISION='pr')` using the binary model `ksi_hya_B.hom` obtained in the previous step.

Keep the file `ksi_hya_B.hom` obtained after this last step as `ksi_hya_ini_B.hom`. It is this model that will be used to initialise the subsequent calculations.
In the `ksi_hya.don` file, indicate the final age: (AGEMAX = 400.d0) and launch the calculations: `cesam2k.out` using `ksi_hya_ini_B.hom` to do the initialisation.
In order to calculate models for other values of the parameters (such as age, overshoot, *etc...*), it is only necessary to adjust the `ksi_hya.don` model to each particular case.

2.16.1 Limiting the output

The information which appears *on line* enables one to closely follow the calculations. This information can turn out to be irrelevant and unnecessarily cumbersome if one wants to run the program in batch mode. For this reason, when the logical variable `baratine` from the `mod_donnees` module is set to `.FALSE.`, most of the information on how the calculations are progressing – namely solving the quasi-static equilibrium, reproducing the atmosphere, and calculating the evolution of the chemical composition and the rotation rate – can be diverted to the files `my_model_static, my_model_atmos, my_model_evol` respectively. In order to do this, one needs to write `baratine=.FALSE.` in the `mod_donnees` module, and then to partially recompile this module, *cf.* § 2.18 (Page 37). When such a limitation is only occasional, it is not necessary to interfere with the `mod_donnees` module and redo a compilation; rather, it is sufficient to set up a file called `blabla` in the calculation environment, see § 3.11.1 (Page 51).

2.17 Processing programs

Programs from the `EXPLOIT` subdirectory enable users to process the ASCII or binary files created by CESAM2*k*:

- `calib2k`: Creates a data file `my_model.don` for solar calibration.
- `des2k_ZC`: Plots the evolution of convection zones as a function of time.

[10] Use binary or ASCII $3M_\odot$ initialisation files if available.

- `des2k_abon`: Plots the abundances as a function of mass or radius.
- `des2k_abontc`: Plots the evolution of X, Y, Z in the centre as a function of time.
- `des2k_abonts`: Plots the evolution of X, Y, Z at the surface as a function of time.
- `des2k_bin`: Plots quasi-static variables from a binary file.
- `des2k_hr`: Plots the HR diagram.
- `des2k_opa`: Plots the relative differences between two opacity tables.
- `des2k_osc`: Plots 1, 2 or 3 models as a function of radius or mass with zoom capabilities.
- `des2k_rot`: Makes separate plots for the angular momentum diffusion variables.
- `des2k_rot_ext`: Plots the rotation rate of the outermost layer in km/s or rad/s as a function of time.
- `des2k_vaiss`: Plots the Brunt-Väissälä frequency profile.
- `f037_2k`: Interpolates a model at fixed points in mass coordinates or radius.
- `fichier_vent`: Constructs a file with the wind's chemical composition.

In each of these programs, information concerning their usage is given in the comments.

2.18 Scripts for running and using CESAM2*k*

The following UNIX scripts (which run in a `csh` shell) from the `SCRIPTS` subdirectory facilitate installing and using[11] CESAM2*k*:

- `calib2k_pms`: calibration procedure for a solar model with PMS.
- `calib2k_zams`: calibration procedure for a solar model initialised as a homogeneous ZAMS star.
- `compile2k`: compilation script including optimisation.
- `compile2k-dbg`: compilation script for debug purposes.
- `compile2k-dbg_list`: compilation script for debug purposes, for a set of subroutines from a list.

[11] The compiler options are those of the LINUX LF95 compiler; it may be necessary to adapt these to the compiler being used.

- `compile2k_list`: compilation script including optimisation for a set of subroutines form a list; creates or updates a library.
- `evol2k_pms`: procedure for doing evolution starting from a homogeneous PMS star.
- `evol2k_zams`: procedure for doing evolution starting from a homogeneous ZAMS star.
- `exe2k`: procedure for running a program with optimisation
- `exe2k-dbg`: procedure for running a program using debug.
- `genere_cesam2k-dbg`: script which creates the library and executable `cesam2k-dbg.out` for debug purposes,
- `genere_cesam2k`: script which creates the library and executable `cesam2k.out` for normal use,
- `lib_del_repl`: script which removes then replaces a subroutine in a library.
- `rempl2k_mod`: script which compiles and replaces one or several modules in the library, and creates the executable module `cesam2k.out`.
- `rempl2k_mod-dbg`: script which compiles and replaces one or several modules in the library, and creates the executable module `cesam2k-dbg.out` for debug purposes.
- `makefile`: makefile which can be used to generate the executable `cesam2k.out`.

Chapter 3

Customisation

Even though the executable is created only once, it is possible to externally adjust a certain number of parameters. These adjustments are done on request, through *ad hoc* files which are placed in the subdirectory where the calculations take place. This will only affect models calculated in that environment. In a general way, a customisation in a given directory will either affect a particular model or all of the models which are to be calculated in that directory. CESAM2k will first look to see if there is a customisation for the model to be calculated. If not, it will then look to see if there is one for all of the models of the directory. Examples of such customisation files can be found in the `EXPLOIT` subdirectory of the distribution. One can:

- Choose to have the interactive dialog in a language other than French[1].
- Use an initial mixture different than those which are implemented.
- Modify the abundance ratios of a mixture.
- Define new isotopic ratios.
- Create an ASCII output file containing a different number of quantities in a different order from that which is implemented.
- Use customised settings.
- Draw boxes corresponding to objectives, and adjust the scales of the on-line HR diagram.
- Adjust the dimensions of the frames and modify the name of the device which is to be used.
- Use data files from an older version of CESAM.
- Divert on-line information onto ASCII files.

As a safety precaution and as explained in each particular case, some customisation files concerning the chemical composition are only taken into account if the `modif_chim` parameter from the `nl_chim` NAMELIST from the data file is `.TRUE.`, *cf.* § 2.7 (Page 26).

[1] Only English in the present version.

3.1 Language

Alternatives for having the interactive dialog in a language other than French are:

1. If one **always** wants comments in a language other than French, then he or she must comment or delete the following instructions in the `cesam` program (towards lines 270-278) in the `SOURCE` subdirectory, **before** compilation:

```
      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
```

   and uncomment the instruction on the following line (279):

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

   Comments will be in English for all subsequent uses.

2. If one only wants the comments in English from time to time, he or she must set up a file called `langue` in the environment, containing the following lines:

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

   Comments will be in English for all subsequent uses in that environment.

By default, *i.e.* when neither of the preceding measures is used, comments will be in French.

PB `Only part of the main comments have been translated into English in the present version of` CESAM2*k*.

PB **Call for man power**: `it is easy to transpose into another language the algorithms set up for working in English or to complete/improve their syntax; volunteers are welcome.`

3.2 Mixture

In order to use an initial mixture different[2] from those that are implemented, write `nom_abon='mixture'` in the data file and place a file called `mixture` containing the chosen mixture (in DeX) in the environment. CESAM2*k* ***refuses*** to work with a mixture in which the normalisation is different from H=12. The following example can be found in the `EXPLOIT` subdirectory:

[2] Also see § 4.2 (Page 54).

```
&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
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
/
```

One can find in the `abon_ini` subroutine the order and identification of elements based on their indices, which are in fact those of the OPAL opacities. Subsequently, when using the option `nom_abon='mixture'`, ***all*** models calculated in that directory will have the initial abundances given in the file `mixture`. Another possibility, which enables further customisation, is to create a file, the name of which is made up of the model's generic name and the extension `.mix`, for example `my_model.mix`. Then, ***only*** the model `my_model` will have that initial chemical composition. When using the option `mixture`, CESAM2k looks to use ***first of all*** the file `my_model.mix`; when this file doesn't exist, CESAM2k uses the file `mixture`; calculations will only proceed if one of the two files are present in the environment.
This option requires setting `MODIF_CHIM=.TRUE.`, *cf.* § 2.7 (Page 26).

PB `When using the option nom_abon='mixture', the relative metal abundances should in principle be the same as those used in the opacities, or at least, for lack of anything better, be taken into account in the opacity subroutine.`

3.3 Isotopic ratios

In order to use isotopic ratios which are different than the default values in CESAM, put a file called `my_model.rap_iso` containing the modified ratios in the calculation environment. If ***all*** the models in the environment need to be calculated with these ratios, rename the preceding file `rap_iso`. For example:

```
&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
/
```

The names of the variables resemble the notations used for isotopic ratios: `be7sbe9` is the isotopic ratio ^{7}Be/^{9}Be, `he3she4z` is the isotopic ratio ^{3}He/^{4}He on the ZAMS once the initial deuterium has been transformed into ^{3}He, `be7sz` is the abundance, which is nearly zero, of the initial ^{7}Be in Z.

As a safety precaution, this file will only be taken into account if the `modif_chim` parameter in the `nl_chim` NAMELIST from the data file is set to `.TRUE.` *cf.* § 2.7 (Page 26).

3.4 Customising abundance ratios

Once the mixture has been defined, for example `solaire_gn` (Grevesse & Noels, 1993), it is possible to modify its abundances. In order to do this, create, in the calculation environment, a file called `my_model.modif_mix` which contains the modifications in DeX to be carried out. If all the models need to be calculated with these modifications, rename the preceding file `modif_mix`. Example from the `EXPLOIT` subdirectory:

```
&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
/
```

The notations resemble the modifications; hence `add_Li=2.0` is the number of DeX by which to increase the lithium abundance, `add_Z=-1.d0` means that the abundances ***of all of the metals*** will be reduced by one DeX. ***As a safety precaution***, this file will only be taken into account if the `modif_chim` parameter in the `nl_chim` NAMELIST from the data file is set to `.TRUE.` *cf.* § 2.3 (Page 20).

If the option `GARDE_XISH` is set to `.FALSE.` in the data file *cf.* § 2.7.1 (Page 28), the parameter `add_Z`, which acts on all the metals, does not have any effect on the metal/Z ratios. Conversely, a non-zero value for `add_C`, for example, will affect the metal/Z ratios.

3.5 Customising the wind's chemical composition

In the standard case, when `mdot` is non-zero, *cf.* § 2.3 (Page 20), the chemical composition of the mass which is gained or lost through wind is that of the model's outermost layer. It is possible to customise this chemical composition so that it is different from that of the outermost layer. In order to do this, put a file called `my_model.vent` in the calculation environment, in which will be given the mass fractions of each chemical element of the mass gain/loss. If all the models in the environment are to use these modifications, rename the preceding file `vent`. An example can be found in the `EXPLOIT` subdirectory:

```
&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
/
```

As the notation suggests, `vt_Li=1.033e-8` is the lithium mass fraction in the wind. Since renormalisation occurs during each use, it is not necessary to normalise the mass fractions in this file. Some of them can be zero or even negative. Matter, whether it is added (`MDOT > 0`) or removed (`MDOT < 0`), is assumed to come from the external convection zone which always exists. The file `vent` can be constructed[3] using the program `fichier_vent` in the `EXPLOIT` subdirectory.

As a safety precaution, this file will be taken into account only if the `modif_chim` parameter in the `nl_fdon` NAMELIST from the data file is set to `.TRUE.` *cf.* § 2.7 (Page 26).

3.6 Fall of planetoids

During a star's evolution, CESAM2k can simulate the fall of planetoids over a limited period of time. In order to do this, put a file called `my_model.planet` in the calculation environment, in which is indicated the mass fraction of each element making up the planetoids' chemical composition, the total number of planetoids to fall onto the star, the interval of time during which the falls occur and the profile giving the intensity of the falls. If all of the models need to be calculated with these modifications, rename the preceding file `planet`. An example can be found in the `EXPLOIT` subdirectory:

```
&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
/
```

The notations resemble what they mean. When `vt_H` and `vt_He` are ***not zero***, non-zero values for `ypl` and/or `zpl` allows one to modify the mass fractions of hydrogen `X`, helium `Y` and metals `Z` of the planetoids. In the example above, the mass abundance of helium will be $Y = 0$, that of metals $Z = 0.9$ and that of hydrogen $X = 1.0 - 0.0 - 0.9 = 0.1$. `age_deb` and `age_fin` are the starting and finishing ages in millions of years, respectively, of the falls. `r_giration` is the gyration radius in AU and `t_giration` the gyration time in years. CESAM2k estimates the angular velocity of planetoids by assuming these follow an arc measuring $\pi/2$ during one gyration time. A negative value for the gyration time corresponds to a reduction of the angular momentum, *i.e.* it is retrograde. If the gyration time is zero, the angular momentum is not modified.
In the subroutine `planetoides.f`, CESAM2k allows 4 different profiles:

1. `rectangle`: The intensity of the falls is constant.

[3]The chemical composition in the example is the meteoritic mixture from Grevesse & Sauval (1998). Furthermore, it is not necessary, this time, to worry about upper/lowercase.

2. `triangle`: The intensity profile of the falls is triangular, in which the maximum is in the middle of the time interval.

3. `parabole`: The intensity profile of the falls is parabolic, in which the maximum is in the middle of the time interval.

4. `gauss`: The intensity profile of the falls is Gaussian and is cut off by the time interval. The maximum is centred in the time interval. The standard deviation is arbitrarily set to one third of the time interval.

This option requires writing `MODIF_CHIM=.TRUE`, *cf.* § 2.7 (Page 26).

3.7 Ascii

In order to create a customised output ASCII file, write `nom_output='ascii'` in the data file and place a file called `sortie_ascii`[4] in the environment, in which are defined the parameters which create the desired structure. The comments in the `output.f` subroutine explain how to write `sortie_ascii`, a ***delicate*** process. The variables' indices are identified in the `ascii.f` subroutine from the `SOURCE` subdirectory. In the following example:

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

the different quantities have the following meanings:

- 12 3 $\longrightarrow$ the output file will contain 12 global variables and 3 local variables,
- 1 2 3 4 5 6 9 10 11 12 13 15 $\longrightarrow$ indices of the 12 global variables (M_star $\times$ $M_\odot$, R_tot $\times$ $R_\odot$, L_tot $\times$ $L_\odot$, z0, x0, alpha, X in CZ, Y in CZ, d2p, d2ro, age, vsal, initial w_rot).
- 1 2 3 $\longrightarrow$ indices of the 3 local variables (Radius, log(M/Mtot), Temperature).
- .FALSE. $\longrightarrow$ the mass will be given as a fraction of the total mass, and not in DeX.
- .TRUE. $\longrightarrow$ the tables start at the centre and go to the surface.
- .TRUE. $\longrightarrow$ the local chemical composition will be added on after the local variables.
- "Fichier pour sortie ascii": text to be used as a header in the output file.

[4] An example can be found in the `EXPLOIT` subdirectory.

The name of the output file is made from the data file's generic name: if the data file is called `my_model.don`, the output file will be `my_model-ascii`. Using the above example, one typically obtains:

```
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
 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
```

Just like for initial abundances *cf.* § 3.2 (Page 40), it is possible to create this customised output file for one particular model, by renaming the file which specifies the output with the name of the model and the extension `.ascii`, for example `my_model.ascii`. If such a file is present in the environment, then its parameters will be the ones taken into account for ***the*** model named `my_model`. If such a file is lacking, CESAM2k will look for the file `sortie_ascii`. If neither file is present, there will be no ASCII output.

3.8 Settings

The settings, which are defined in the program `cesam.f` based on the `PRECISION` parameter in the `my_model.don` data file, can be customised by setting `PRECISION='rg'` and by putting in the environment a file called `reglages`[5] which contains the necessary adjustments. The following is an example of a `reglages` file:

```
&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,
```

[5] An example can be found in the `EXPLOIT` subdirectory.

```
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,
ctem=15.d0,
cter=0.0d0,
ctet=0.d0,
mvt_dis=.FALSE.,
dn_fixe=0.05d0,
dpsi=0.05d0,
w_form=.05d0,
mu_saha=.FALSE.,
q0=0.1d0,
l0=4,
new_bv=.TRUE.,
iter_qs=0, 0, 3, 0, 0, 0, 7
/
```

The meanings of the different parameters are:

m_qs=2 : Order of the B-splines used for quasi-static variables.

m_ch=3 : Order of the B-splines used in interpolating the chemical composition.

m_rot=3 : Order of the B-splines used in interpolating the angular momentum.

m_tds=2 : Order of the B-splines used in interpolating the gravitational energy.

m_ptm=2 : Order of the B-splines used in interpolating mass loss.

ordre=1 : Order of the integration scheme for the nuclear reactions with `rk_imps`.

precix=1.d-4 : Precision of the Newton-Raphson iterations of the spatial integrations.

precit=0.1d0 : Maximal relative variation for the time integration of the chemical composition.

ro_test=0.1d0 : Test on variation of gravitational energy if `ro > ro_test`.

psi0=0.05d0 : Mesh increment to be maintained.

d_grav=0.5d0 : Maximal temporal variation of gravitational energy.

loc_zc=5.d-4 : Precision in locating radiation/convection zone boundaries.

dtmax=200.d0 : Maximal time step.

ini0=5 : Number of Newton-Raphson iterations with re-estimation of chemical composition, angular velocities and radiation/convection zone boundaries.

n_atm=75 : Number of layers in the atmosphere.

kipp=.TRUE. : Use of Kippenhahn's approximation.

en_masse=.TRUE. : Quasi-static Lagrangian variables.

ctel, ctep, ctem, cter, ctet : Mesh factors.

mvt_dis=.TRUE. : Adjustments of the chemical composition due to the movement of discontinuities..

dn_fixe=0.05d0 : Limit on the rate of variation of the number of layers before needing to readjust the fixed grid for the interpolation of the chemical composition.

dpsi=0.02 : Limit on the rate of variation of the "mesh function" before needing to modify the number of layers.

w_form=0.05 : Coefficient with an absolute value less than one, which allows the modification of the initial rotation profile.

mu_saha=.FALSE. : The program will assume that the medium is totally ionised when calculating the mean molecular weight.

q0=0.1d0 : In the ASCII output files, a point is inserted at $q0 > 0$ times the distance between the first two points.

l0=4 : In the ASCII output files, add l0+1 points between lim_CZ and lim_CZ+1, l0 points between lim_CZ+1 and lim_CZ+2..... 1 point between lim_CZ+l0 and lim_CZ+l0+1 and the symmetrical counterpart between lim_CZ and lim_CZ-l0.

new_bv=.TRUE. : Calculate the Brunt-Väissälä frequency using $\varphi = \frac{\partial \ln \rho}{\partial \ln \mu}$.

iter_qs=0, 0, 3, 0, 0, 0, 7 : Enables one not to take into account a low precision on some of the quasi-static variables.

The adjustment of the mesh factors ***is tricky***. The default values are `ctel=0, ctep=-1, ctem=15, cter=0, ctet=0`. The constants `ctep` and `ctet` should always be ***negative***, since they affect, respectively, the pressure and the temperature, both of which decrease from the centre to the surface.

Just like for initial abundances *cf.* § 3.2 (Page 40), it is possible to further customise these adjustments by renaming the adjustment file with the name of the model and the extension `.rg`, for example `my_model.rg`. If such a file is present in the environment, then its parameters will be the ones which are taken into account for ***the*** model named `my_model`. If such a file is lacking, CESAM2k will use the parameters in the file `reglages`. Otherwise, no calculations will be done.

3.9 Zoom

It is possible to customise the on-line HR diagram by drawing a rectangular target for the evolutionary path, as well as by adjusting the range of the axes so as to centre them around the target value and the surrounding area, the dimensions of which are given in DeX of $\log T_{\rm eff}$ and $\log L$. One can also draw a target for [Fe/H] in the frame which shows the abundances as a function of mass. In order to do any of these, write `nom_des='zoom'` in the data file and put a file called `zoom` with the necessary data in the environment. The following is an example of such a file[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
/
```

- There will be no customisation if `teff_des` and/or `l_des` is negative or zero.
- The zoom will take effect only if `zoom_t` and `zoom_l` are non-zero.
- To avoid plotting a target for metallicity, write `fesh_des=100`.
- The maximal (minimal) `logteff` and `logl` values, *i.e.* `logteff_max (min)`, `logl_max (min)`, ***will override*** those of the target; in the preceding example, the ranges for the logarithm of the effective temperature and that of the luminosity will be respectively $[4.6, 3.5]$ and $[2.5, 5.]$.
- If one **only** wants to adjust the range of the axes in the HR diagram, he or she should write something like:

```
&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.,
```

[6]This example can be found in the `EXPLOIT` subdirectory

```
dfesh_des=0., 0.,
logteff_max=3.6,
logteff_min=2.5,
logl_max=2.,
logl_min=-2.5
/
```

- If one **only** wants to centre the HR diagram around the target, he or she should write:

```
&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.
/
```

The meanings of these variables are:

teff_des=6530. : Effective temperature at the centre of the target.

dteff_des=50., -30. : Margins on the effective temperature.

zoom_t=0.2, -0.1 : Size of the area around the target in DeX of the effective temperature.

l_des=6.844 : Luminosity at the centre of the target.

dl_des=0.6, -0.4 : Margins on the luminosity.

zoom_l=0.1, -0.2 : Size of the area around the target in DeX of the luminosity.

fesh_des=0.05 : Metallicity target.

dfesh_des=0.03, -0.04 : Margins around the metallicity target.

logteff_max=4.6 : $\log_{10}$ of the maximal effective temperature in the HR diagram.

logteff_min=3.5 : $\log_{10}$ of the minimal effective temperature in the HR diagram.

logl_max=5. : $\log_{10}$ of the maximal luminosity in the HR diagram.

logl_min=2.5 : $\log_{10}$ of the minimal luminosity in the HR diagram.

Is possible to do further customisation, by renaming the `zoom` file with the name of the model and the extension `.zoom`, for example `my_model.zoom`. The previous measures will then apply ***only to*** to the model `my_model`. When the `zoom` option is set, CESAM will first seek to use the file `my_model.zoom`. If this file doesn't exist, then the `zoom` file is used.

3.10 Device

In order to adjust the dimensions of the frames and the name of the graphical device, put a file called `device` in the calculation environment which will contain the necessary data. The following is an example[7] of such a file:

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

The meaning of these variables are (the units are cm):

h=6.5 : Height of the frames.

dh=1.5 : Vertical space between the frames.

ld=8.5 : Width of the frames.

dl=2. : Horizontal space between the frames.

xleft=1.8 : Distance between the plot and the left side of the screen.

ybot=1.4 : Distance between the plot and the bottom of the screen.

device='/xw' : Name of the graphical device.

These values can be modified directly in the `mod_donnees` module in the `SOURCE` subdirectory.

3.11 Physique45

Placing a file called `physique45` in the calculation environment *cf.* § 2.15 (Page 35) enables one to use data files from CESAM4 or CESAM5 with non-default physics settings. For example:

[7] This example can be found in the `EXPLOIT` subdirectory

```
&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_DIFFW='diffw_mpz',
 NOM_THW='cons_loc_mnt_cin',
 NOM_PERTW='pertw_0',
 NOM_ETAT='etat_ceff'
 NOM_OPA='opa_yveline'
 NOM_NUC='ppcno3ac10'
 NOM_NUC_CPL='Cau-Fow'
 NOM_ATM='lim_atm'
 NOM_TDETAU='roger_00'
/
```

This file can be found in the `EXPLOIT` subdirectory.

3.11.1 Limiting the output

The information which appears *on line* enables one to closely follow the calculations. This information can turn out to be irrelevant and unnecessarily cumbersome if one wants to run the program in batch mode, *cf.* § 2.16.1 (Page 36). When such a limitation is only occasional, it is sufficient to set up a file called `blabla` in the calculation environment:

```
&NL_BLABLA
baratine=.FALSE.
/
```

This file can be found in the `EXPLOIT` subdirectory.

Chapter 4

Extensions

4.1 Adding a chain of thermonuclear reactions

When implementing a new chain of thermonuclear reactions, it is usually necessary to add several isotopes as well as several thermonuclear reactions.

- ***To add an isotope***[1]:
  1. In the `mod_nuc` module, increase the `PARAMETER niso_tot` by one, for example: `niso_tot=28`.
  2. In the `taux_nuc` subroutine:
     (a) In the declarations, add the isotope's mass excess and symbol, for example: `Fe56=-60.6054d0`.
     (b) Add the isotope's mass, (it is recommended to add the isotope's mass in the `ctes_85`, `ctes_94` and `ctes_94m` subroutines which contain physical constants, and subsequently in the declarations of the `mod_donnees` module and in the restriction clause "`ONLY`" in the `taux_nuc` subroutine, for example: `nucleot(27)=afe56, afe56=55.847d0, afe56, af18...`).
     (c) Add the isotope's electric charge, for example: `zit(28)=11`.
     (d) Add the name of the isotope, for example: `'B11 '`. (the contents of these 4 character strings should be flush left unless the symbol contains 2 characters, in which case it should be centred.).
  3. If need be, in the `abon_ini` subroutine:
     (a) Add an isotopic ratio for this new isotope:
        i. Introduce its value in the list at the end of the subroutine.
        ii. Indicate its name in the `nl_rap_iso` and `nl_modif_mix` NAMELISTs in the part of the subroutine with declarations.
  4. If need be, in the `saha` subroutine, add the ionisation potential for this new element

[1] By default, CESAM2k allows the possibility of working with the chemical elements ranging from hydrogen ($Z = 1$) to nickel ($Z = 28$). In order to work with heavier elements, one needs to add these by putting them in the `abon_ini` subroutine using a procedure similar to the one described here for adding isotopes.

- ***To add a reaction***:

  1. In the `taux_nuc` subroutine:
     (a) Complete the list of comments by including the symbol which represents the reaction and its index, for example :: `réaction 46: B11(p,g)C12`.
     (b) Initialise the reaction's name, the mass deficit, and the electric charges of the relevant nuclei, for example: `nom_react(2)='H2(p,g)He3'`
     `nuc=H2+p-He3` ; `qt(2)=nuc` ; `izzt(2,1)=1` ; `izzt(2,2)=1`.
     (c) Enter the natural logarithm (ln) of the reaction rate[2] `rt(i)` and do not forget the 1! or 2! or 3! in the denominator, for example: `réaction 8: C12(p,g)N13(e+ nu)C13 z0=6, z1=1.....` If need be, recalculate the coefficients using $S(0)$, $S'(0)$, $S"(0)$ (Lang, 1968, eq. 4-49).
  2. Write the subroutine for the thermonuclear reactions using one of the other subroutines, such as `ppcno12BeBFe.f`, as an example.
  3. In the `tabul_nuc` subroutine:
     (a) Add the new chain, for example: `CASE('ppcno12')`
     (b) Enter the parameters needed for calculating the rates, the number of reactions, the correspondences between indices, the isotopes which intervene ..., for example: `nreac=30 ; ind(16)=31 ; WRITE(2,6) ; WRITE(*,6)....`
  4. In the `mod_nuc` module:
     (a) Increase `PARAMETER nreac_tot` by one, for example: `nreac_tot=46`
     (b) Insert an `INCLUDE` statement with the name of the subroutine which implements the chain of reactions, for example: `INCLUDE 'ppcno9.f'`
  5. In the generic subroutine `nuc`:
     (a) Make it active and have it recognise the new subroutine.
  6. Test the algorithms which have been modified by using the programs in the `TEST` subdirectory, `test_tabul_reac.f` to check the reaction rates and `test_jacobien_reac_nuc.f` to insure that the derivatives are accurate.

4.2 Adding an initial mixture

As has been described in § 3.2 (Page 40), it is possible to customise the initial abundances using a file called `mixture`. However, if this mixture is of general interest[3],

[2] For historical reasons, these calculations are done using natural logarithms. The reasons were to avoid truncation errors on the one hand, and numerical overflows due to the limitation at 10^{38} on the other hand.

[3] Please communicate the modified subroutine to those who keep CESAM2k up to date, so that the community may benefit from your work.

it is better and very easy to add its table of abundances directly in the source, in the `abon_ini` subroutine, giving it a name with up to 20 characters. To remake the executable module, one just needs to apply the `rempl2k_mod` script to the `mod_nuc` module: `rempl2k_mod mod_nuc`.

Chapter 5

Debug

5.1 Debug

Since CESAM2k's structure is based on modules, the compiler can efficiently check the syntax. On the down side, when making improvements and/or debugging it is necessary to recompile the entire module which contains the subroutine which is being worked on. When installing CESAM2k, it is useful to create a debug library `libcesam2k-dbg.a` as well as the main library `libcesam2k.a`. This library is automatically created when running the `genere_cesam2k-dbg` script from the `SCRIPTS` subdirectory. In order to do this, use the compiler's debug options; the `compile2k-dbg` procedure form the `SCRIPTS` subdirectory is provided as an example. Constructing the debug library is around 5 to 10 times faster than creating the main library but running it is also 5 to 10 times slower.

Example: a bug has been detected in the `des_m` subroutine from the `mod_cesam` module[1]. We introduce debug instructions in `des_m`. In the the `TESTS` subdirectory, we run the program `test_cesam` in which we have included the `mod_cesam` module:

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

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

PROGRAM test_cesam

USE mod_cesam

IMPLICIT NONE

CALL cesam

STOP

END PROGRAM test_cesam
```

[1] *Good* compilers will give you the name of the subroutine in which they have found a bug.

When the compiler links the object files together, the `mod_cesam.mod` file created in the `TESTS` subdirectory will be used rather than the one from the `SOURCE` subdirectory. During the execution, the debug instructions, which have now been taken into account, will help to detect the bug. Once the `des_m` subroutine has been corrected, running the `mod_repl mod_cesam` script in the `SOURCE` subdirectory will replace the `mod_cesam` module in the library and recreate the executable `cesam2k.out`. As a result, none of the files from the `SOURCE` subdirectory will have been moved.

Appendix A

Installation under LINUX

A.1 Creating the library using `ifc`

The F95 Standard, such as it has been defined in 1996, is strictly followed by the `ifc` compiler[1]. Since then, we noticed a "stupid" limitation in this Standard which does not allow putting an `ALLOCATABLE` array whereas:

```
REAL(kind=dp), INTENT(out), POINTER, DIMENSION(:)
```

is allowed. In certain contexts, `POINTER` and `ALLOCATABLE` are synonymous. Some time later, a Technical Report correcting and removing this limitation came out. The Compaq/Digital (Unix workstation, Windows PC) Lahey-Fujitsu (Linux, Windows) compilers have been corrected; INTEL only seems to have made the correction in the last version of `ifc`.

Remedy: Replace `ALLOCATABLE` by `POINTER`, each time the compiler gives an error message. Later on, with the up-to-date version of `ifc`, the problem should sort itself out. Currently, the Fortran 2003 Standard distinguishes between `ALLOCATABLE` and `POINTER`.

The corrections are as follows:

1. In the `tabul_nuc.f` subroutine, transform:

```
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
```

   into:

```
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. In the `tabul_nuc` subroutine, transform:

[1] Diagnostic, comments and remedies by B. Pichon.

```
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
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
```

into:

```
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. In the `mod_donnees` subroutine, transform:

```
CHARACTER (len=4), SAVE, PUBLIC, ALLOCATABLE, DIMENSION(:) :: nom_elem
```

into:

```
CHARACTER (len=4), SAVE, PUBLIC, POINTER, DIMENSION(:) :: nom_elem
```

A.2 Installation of PGPLOT

A.2.1 Complete installation with `ifc`

Thierry Corbard, from the Observatoire de la Côte d'Azur, has written a PGPLOT configuration file `ifc_gcc.conf` under LINUX for the INTEL's F90 compiler `ifc`. In order to create PGPLOT's source using this compiler:

1. Go into PGPLOT's source subdirectory `/usr/local/src/pgplot/sys_linux`.

(a) Add the configuration file: `ifc_gcc.conf`, which can be found in the `SCRIPTS` subdirectory.

(b) Create a subdirectory called `ifc_src`: `mkdir ifc_src.`

(c) Put, into this subdirectory, the file `ifc_src/grgenv.f` which can be found in the `SCRIPTS` subdirectory

2. Run:
`/usr/local/src/pgplot/makemake /usr/local/src/pgplot linux ifc_gcc`

A.2.2 Installation without tears

Under LINUX, it is possible to install PGPLOT without too much difficulty by proceeding in the following manner[2]:

1. Using a search engine such as Google, look for `pgplot rpm`.

2. Choose, among the different possibilities, the one that allows you to download a recent version which corresponds to your processor and operating system. For example, `pgplot-5.2.2-1.i686.rpm` will do for a Pentium4 under Mandrake[3].

3. Download this software. Once the download has finished, go into the directory where the software was downloaded. Click on the name of the software, get root privileges, and the software will install itself.

4. The system places two shared libraries, `libpgplot.so.5.2.0` and `libpgplot.so.5`, in the `/usr/lib` directory and creates a subdirectory called `/usr/lib/pgplot`. The subdirectory contains 3 files: `grexec.f, rgb.txt, grfont.dat.`

5. In the `pgplot` subdirectory, create two links to the shared libraries:

   `ln -s /usr/lib/libpgplot.so.5 libpgplot.so,`

   `ln -s /usr/lib/libpgplot.so.5.0 libpgplot.so.0`

6. In the `.cshrc` file, add:

   (a) The path to `PGPLOT`: `LD_LIBRARY_PATH **** :/usr/lib/pgplot`.

   (b) The environment variable: `setenv PGPLOT_DIR /usr/lib/pgplot`.

A.2.3 CESAM2*k* without PGPLOT

It is possible to use CESAM2*k* without `PGPLOT`. In order to do this, uncomment the last line of the `MOD_CESAM` module before making the executable module. The file `pgplot_factice.f` contains emulations of the subroutines in `PGPLOT`. Obviously, write `NOM_DES='no_des'` and `NOM_DES_ROT='no_des'` in the data files; avoid trying to link with the `PGPLOT` library; also, do not indicate the path to this library in the `.cshrc` file, or its equivalent.

[2]This procedure has been tested on the Mandrake and SuSE LINUX distributions.
[3]This software can be found in the `SCRIPTS` subdirectory of the distribution.

A.3 Example of a MAKEFILE

The following `MAKEFILE`, written by Th. Corbard O.C.A., can be found in the `EXPLOIT` subdirectory. The parameters correspond to INTEL's `ifc` compiler and the paths to a standard implementation. Its use is described in § 1.5.1 (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

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\
        delete_doubles.f\
        difdiv.f\
        fermi_dirac.f\
        gauss_band.f\
        horner.f\
        intgauss.f\
        linf.f\
        matinv.f\
max_local.f\
        min_max.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.$(mod_ext): \
        ctes_85.f\
        ctes_94.f\
ctes_94m.f\
        ini_ctes.f\
        lit_nl.f\
        print_ctes.f\
        mod_kind.$(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.$(mod_ext): \
         kappa_cond.f\
         opa.f\
 opa_compton.f\
         opa_gong.f\
         opa_houdek9.f\
         opa_int_zsx.f\
         opa_opalCO.f\
         opa_opal2.f\
         opa_yveline.f\
         opa_yveline_lisse.f\
         z14xcotrin21.f\
         mod_kind.$(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_rota3.f\
        coeff_rota4.f\
        collision.f\
        coulomb.f\
        diffm.f\
        diffm_br.f\
        diffm_mp.f\
        difft.f\
        difft_gab.f\
        difft_nu.f\
        difft_sun.f\
        diffus.f\
        diffw.f\
        diffw_mpz.f\
        diffw_p03.f\
        ecrit_rota.f\
```

```
          eq_diff_chim.f\
          eq_diff_poisson.f\
          eq_diff_rota3.f\
          eq_diff_rota4.f\
          evol.f\
          f_rad.f\
          initialise_poisson.f\
          initialise_rota.f\
          initialise_u.f\
          initialise_w.f\
          lmix.f\
          pertw.f\
          pertw_loc.f\
          pertw_ptm.f\
          pertw_sch.f\
          resout_chim.f\
          resout_rota3.f\
          resout_rota4.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.$(mod_ext): \
         add_ascii.f\
         ascii.f\
         cesam.f\
         des.f\
         des_m.f\
         des_r.f\
         dnunl.f\
         list.f\
         output.f\
         osc_adia.f\
         osc_invers.f\
         osc_nadia.f\
         mod_kind.$(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\
 ctes_94m.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)
```

Appendix B

List of ASCII files

For a description of the ASCII data file `my_model.don`, see § 2.3 (Page 20).

B.1 Files for oscillations

The ASCII files used for the calculation of stellar oscillations have a common header. Their contents differ according to their type.

`header`:

- Lines 1-4: identification and specification of which physics is used.
- Line 5: number and name of the chemical elements which are used.
- Line 6: number of layers, of "global" parameters for the model (13), of variables, of chemical elements (redundant), angular velocity index (if there is no rotation, this index is -1).

Example:

```
Fichier pour inversion: test-inv.osc
CESAM2k version 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
```

B.1.1 File for adiabatic oscillations

```
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)=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
```

B.1.2 File for non-adiabatic oscillations

```
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)=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
```

B.1.3 File for 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)=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
```

B.1.4 Using the maximum number of layers

With certain types of precisions, *cf.* § 2.3 (Page 20), it is possible to create an ASCII file for oscillations with at least a predetermined number of layers, the maximum number of layers used when calculating the last models. In order for this to occur, the following criteria must be satisfied:

- Uses one of two types of precision: ‘sa’ or ‘co’.
- ***AND*** fulfils one of the following criteria:
  - Age of the model one million years less than the maximal age.
  - Number of the model equal to `NMAX_MODELS` – 1.
  - Effective temperature close to the limit set by the user.
  - Central hydrogen abundance close to the limit set by the user.
  - Size of the helium nucleus close to the limit set by the user.

B.2 Files for the HR diagram

The file for plotting/using the HR diagram is systematically implemented during each run of CESAM2k. It is created when a model is initialised on the homogeneous zero age main sequence or on the homogenous pre-main sequence. It is completed when continuing the evolution of a previous model.

For each model, the first line contains:

1. the age in 10^6 years;
2. the number of elements in the chemical composition vector;
3. the number of radiation/convection zone boundaries,
4. the number of the model,
5. for each radiation/convection zone boundaries, the letter "T" or "F" depending on whether it is the beginning or end of a convection zone;
6. depending on the number of boundaries, one or several lines which indicate:
   (a) $\log T_{\rm eff}$,
   (b) $\log L/L_\odot$,
   (c) $\log R/R_\odot$
   (d) for each boundary, $M_\star/M_\odot$, $(M_\star - m)/M_\odot$, $r/R_\odot$, $r_{\rm ov}/R_\odot$, ($r_{\rm ov}$ being the radius of the boundary after it has been extended by overshoot).

The following conventions are used for particular cases:

– totally convective model: (only) 1 boundary "F" which is placed at the centre ($r/R_\odot = 0$),

– totally radiative model: 0 boundaries "F", placed at -100 ($r/R_\odot = -100$).

This is followed by the names of the chemical elements and their abundances per mass unit, at the centre and at the surface. If there is no overshoot, the values $r_{\rm ov}$ of the radii of the extended boundaries are set to -100. If diffusion of the angular momentum is taken into account, an extra line, starting with `Wrot`, gives the linear and angular velocities of the outermost layer. For example:

```
 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  <-- only with diffusion of the angular momentum
 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  <-- only with diffusion of the angular momentum
................................

 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  <-- only with diffusion of the angular momentum
```

The files for the HR diagram are read by the `lit_hr` subroutine in the `mod_exploit` module.

B.3 ASCII files for the variables relating to the diffusion of angular momentum

The ASCII file for the variables relating to the diffusion of angular momentum is created in the calculation environment depending on circumstances described in § 2.10 (Page 30). It can be processed using the plot program `des2k_rot` in the `EXPLOIT` subdirectory, *cf.* § 1.3 (Page 2).

`header`:

- Lines 1-4: identification of the model and specification of which physics is used.

- Line 5: names of the variables in the order in which they are given in the file.
- Line 6: number and names of the chemical elements which are used.
- Line 7: number of layers, of variables, of chemical elements (redundant), identifier which indicates which equations were used to represent the diffusion of angular momentum (3 for Talon & Zahn 1997, 4 for Mathis & Zahn 2004), index of ^{4}He, number of the model, index of ^{7}Li.
- Following lines: `FORMAT(5es19.12)` variables and chemical composition.

The model starts from the centre and goes to the surface. Example:

```
Fichier pour la diffusion du moment cinétique: 1.0_coeff_rota.dat
CESAM2k version V2.2.0 lagr colloc  2  3 pr no diffus, 18 Mai 2006 11h59
Physique utilisée: etat_eff, opa_yveline, conv_jmj, ppcno9, ctes_94, NACRE, diff_tz97
solaire_gn, lim_atm, hopf, pertm_ext, pertw_0, diffm_mp, difft_nu, diffw_mpz, diff_tz97
Variables : R, M, Omega, U, Theta, Psi, Lambda, Flux, Deff, Dh, Dv, T, ro, grad_mu, Xchim
 10  H1  He3  He4  C12  C13  N14  N15  O16  O17  Si28
       601        24        14        10         3         3        42         0
 0.000000000000E+00 0.000000000000E+00 2.492796313373E-06 1.222009447047E-15-2.215093172271E-11
 4.146564017724E-13 2.345111261140E-11-9.012551719834E-35 2.621399230830E+00 3.397109673134E+00
 2.621399230830E+00 1.340098717163E+07 8.330731227990E+01 1.426753865039E-02 6.871377248045E-01
 5.018205056432E-05 2.922381643859E-01 1.473316033117E-05 4.025153911526E-06 5.082420872692E-03
 2.299827930187E-07 9.638362729473E-03 6.971562756039E-06 5.827185297122E-03
 1.644625353770E-02 2.636507234666E-04 2.492796313373E-06 1.222009447047E-15-2.215093172271E-11
 4.146564017724E-13 2.345111261140E-11-9.012551719834E-35 2.621399230830E+00 3.397109673134E+00
 2.621399230830E+00 1.340098717163E+07 8.330731227990E+01 1.426753865039E-02 6.871377248045E-01
 5.018205056432E-05 2.922381643859E-01 1.473316033117E-05 4.025153911526E-06 5.082420872692E-03
 2.299827930187E-07 9.638362729473E-03 6.971562756039E-06 5.827185297122E-03
 2.464852077382E-02 8.840621368772E-04 2.492616591912E-06 3.215689566074E-15-4.144404321209E-11
 1.675631823126E-12 4.481166434756E-11-5.704344611195E-34 2.614052191228E+00 3.746479760886E+01
 2.614052191228E+00 1.334650928015E+07 8.276105949309E+01 8.706470878900E-03 6.872913782734E-01
 5.143754364837E-05 2.920832735684E-01 1.464390266467E-05 4.002793748349E-06 5.082554714526E-03
 2.301323508453E-07 9.638504967128E-03 6.795246250354E-06 5.827178857934E-03

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

 8.794865539306E-01 9.999999998874E-01 2.361680091400E-06-5.768386555867E-05 2.367155329527-282
-6.926405852191-289 3.012743146671-282-1.572130233500E-28 1.000000000000E+13 1.000000000000E+13
 1.000000000000E+13 9.146301617794E+03 4.727352051188E-07 0.000000000000E+00 6.999999992655E-01
 8.827001090114E-05 2.799117307236E-01 3.424868374763E-03 4.127760067609E-05 1.059003261745E-03
 4.167448298824E-06 9.640129747024E-03 3.902528976547E-06 5.826651038552E-03
 8.795031344698E-01 1.000000000000E+00 2.361680091400E-06-5.770215872484E-05 6.993868019057-283
-2.308801950730-289 1.022180710478-282-1.515562617419E-28 1.000000000000E+13 1.000000000000E+13
 1.000000000000E+13 8.883218491357E+03 4.555466484484E-07 0.000000000000E+00 6.999999992655E-01
 8.827001090114E-05 2.799117307236E-01 3.424868374763E-03 4.127760067609E-05 1.059003261745E-03
 4.167448298824E-06 9.640129747024E-03 3.902528976547E-06 5.826651038552E-03
```

Bibliography

Adelberger, E., et al.: Solar Fusion Rates. Rev. Mod. Phys. **70**, 1265–1291 (1998)

Anders, E., Grevesse, N.: Abundances of the elements - Meteoritic and solar. Geochimica et Cosmochimica Acta **53**, 197–214 (1989)

Angulo, C., Arnould, M., Rayet, M., and the NACRE collaboration: A compilation of charged–particule induced thermonuclear reaction rates. Nuclear Physics A **656**, 3–183 (1999)

Böhm-Vitense, E.: Über die Wasserstoffkonvektionszone in Sternen verschiedener Effektivtemperaturen und Leuchtkräfte. Zs. f. Astrophys. **46**, 108 (1958)

Canuto, V.M., Mazitelli, I.: Stellar turbulent convection - A new model and applications. ApJ **370**, 295–311 (1991)

Caughlan, G.R., Fowler, W.A.: Thermonuclear Reaction Rates V. Atomic Data and Nuclear Data Tables **40**, 284–334 (1988)

Cox, J.P., Giuli, R.T.: Principles of Stellar Structure, vol I & II. Gordon & Bretch, New-York, (1968)

Grevesse, N., Noels, A.: Cosmic Abundances of the Elements. In: Prantzos, E. Langioni-flam, M. Casse (eds.) Origin and Evolution of the Elements. Cambridge Univ. Press, 14, (1993)

Grevesse, N., Sauval, A.J.: Standard Solar Composition. Space Science Revue, **85**, 161-174 (1998)

Houdek, G., & Rogl, J.: On the accuracy of opacity interpolation schemes. Bulletin of the Astronomical Soceity of India **24**, 317 (1996)

Iglesias, C.A., & Rogers F.J.: Opacities for the solar radiative interior. ApJ **371**, 408–417 (1991)

Lang, K.R. : Astrophysical formulae, Springer-Verlag Berlin (1968)

Mathis, S., Zahn, J.P.: Transport and mixing in the radiation zones of rotating stars. I. Hydrodynamical processes, A&A **425**, 229–242 (2004)

Mathis, S., Palacios, A., Zahn, J.P.: On shear-induced turbulence in rotating stars, A&A **425**, 243–247 (2004)

Mihalas D., Dappen, W., Hummer, D.G.: The equation of state for stellar envelopes. II - Algorithm and selected results. ApJ **331**, 815–825 (1988)

Mitler, H.E.: Thermonuclear ion-electron screening at all densities. I - Static solution. ApJ **212**, 513–532 (1997)

Palacios, A., Talon, S., Charbonnel, C., Forestini, M.: Rotational mixing in low mass stars : I Effects of the μ gradient in main sequence and sub giant Pop I stars. A&A **399** 603–617 (2003]

Waldron, W.L.: Stellar winds: Observational evidence for a hot-cool star connection.In NASA. Goddard Space Flight Center The Origin of Nonradiative Heating/Momentum in Hot Stars A.Ap **229**, 95-100 (1985)