instituto de astronomía
unam

NSCool User's guide

The Control Files

Dany Page

Instituto de Astronomía
Universidad Nacional Autónoma de México

Cool_*.in

The Input Master File “Cool_*.in”

The master input file:

Cool_Try.in

```
   'NEW'
BASIC MODEL FILES:
   'EOS/Crust/Crust_EOS_Cat_HZD-NV.dat'
   'EOS/v14/APR_EOS_Cat.dat'
   'TOV/Profile/Prof_APR_Cat_1.4.dat'
OTHER MODEL FILES:
   'I_Files/I_Struct_1.6e14-4e11-1e10_normal.dat'
   'I_Files/I_Bound_Fe.dat'
   'I_Files/I_Pairing_SFB-a-T73.dat'
   'I_Files/I_Neutrino_1.dat'
   'I_Files/I_Conduct_21.dat'
   'I_Files/I_Heat_0.dat'
   'I_Files/I_Bfield_0.dat'
   'I_Files/I_Accretion_0.dat'
OUTPUT FILES:
   'Model_1/I.dat'
   'Model_1/Teff_Try.dat'
   'Model_1/Temp_Try.dat'
   'Model_1/Star_Try.dat'
```

-:-- Cool_Try.in All (20,0) (Fundamental)

Its content is briefly described in the next slides.
For more details see: NSCool_Guide_Control.

It is defined in NSCool.f with:

NSCool.f

```
c ***  Choose between two input: ************************************
C      Ask for the input file ***************************************
       write(6,*) 'Input master file ='
       read(5,*)filename
c ***  Can add here the directory where "Cool_*.in" is:
c       filename='Model_1/'//filename
c ***  Or define it completely here: ********************************
c       filename='Model_1/Cool_Try.in'
c       write(6,*)'Using as input: ',filename
c**********************************************************************
       open(unit=15,file=filename,status='old')
```

-:-- NSCool.f 14% (193,0) (Fortran)

If you’re tired of typing in the directory, uncomment the
`c   filename=‘Model_1’//filename`

Or, if you’re always using the same input file (e.g., while debugging) use the second version (uncomment it, and comment out the first version)

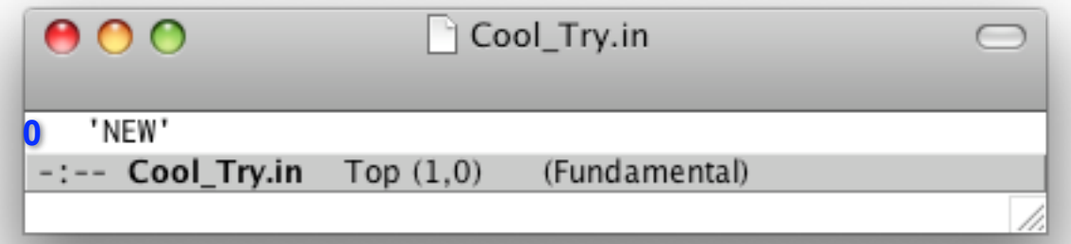
The way it is presently working, NSCool will look for these files as they are defined in the input file.
The way the files are defined in `Cool_Try.in`, since the subdirectories `EOS, TOV, I_Files, Model_1`, ..., are in the `NSCool` directory, one has to run `NSCool` from this directory (and not from the `Code` subdirectory).
But if you define the files as, e.g., ‘`../EOS/Crust/Crust_EOS_Cat_HZD-NV.dat`’, you can run from `Code`.

The input file line by line (1)

0: This specifies how much info there is in the core EOS file. Its possible values are:
'old' , 'new' , 'NEW' , 'QRK' , and 'STR'.

Presently, only 'NEW' and 'STR' work properly !

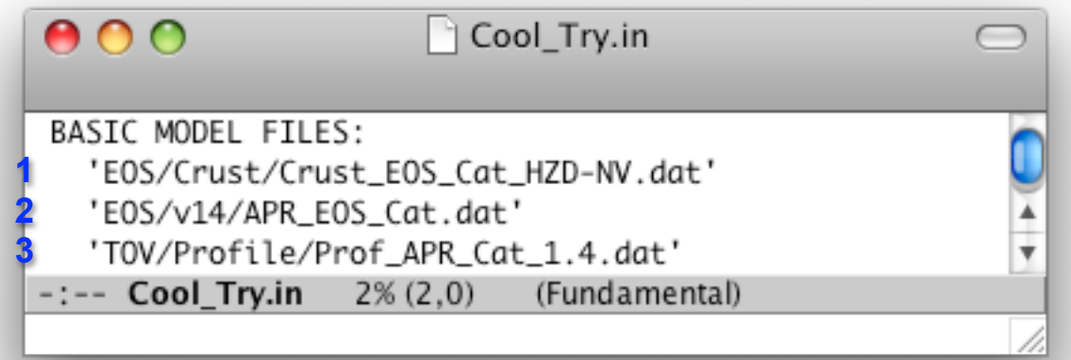
- 'NEW' includes leptons, nucleons and hyperons.
- 'STR' is for strange stars with a thin baryonic crust.
- 'QRK' was as 'new' but with quark matter ("hybrid star"), allowing for a mixed phase. I think it is not anymore compatible with the present version of NSCool, but this should be easy to fix (by making an extension of 'NEW' instead of 'new').



- 'old' and 'new' may work, but this needs to be checked.

This terminology obviously reflects the evolution of NSCool with time !

3: contains the profile of the star as calculated by the TOV integrator. Since TOV is integrated by a Runge-Kutta scheme, during initialization a new grid is defined, by interpolations, more appropriate for NSCool.



1: the crust EOS and
2: the core EOS:they are used to define the "chemical composition" of the star, i.e., abundances of each type of particle, and nuclei in the crust, as well as Fermi momenta, effective masses, etc ... All this is performed during initialization.

The input file line by line (2)

4: contains parameters which define the grid used.
I like to give files a name which is pretty indicative of its content. This name indicates that this specific file defines: $\rho_{crust\text{-}core} = 1.6 \cdot 10^{14}$ gm cm^{-3}, $\rho_{drip} = 4 \cdot 10^{11}$ gm cm^{-3} and $\rho_b = 10^{10}$ gm cm^{-3} (that is the "surface", i.e., the position of the outer boundary), and a "normal" size grid (about 100 grid points in the core and 250 in the crust).

5: defines the outer boundary condition, i.e., the envelope model glued at r_b.
(This particular file defines a heavy element envelope with iron at the surface.)

6: define which pairing gaps are used (for superfluidity/superconductivity).

7: allows some control to turn on or off some neutrino processes.

8: same thing, but for the thermal conductivity.

Cool_Try.in

```
 OTHER MODEL FILES:
4   'I_Files/I_Struct_1.6e14-4e11-1e10_normal.dat'
5   'I_Files/I_Bound_Fe.dat'
6   'I_Files/I_Pairing_SFB-a-T73.dat'
7   'I_Files/I_Neutrino_1.dat'
8   'I_Files/I_Conduct_21.dat'
9   'I_Files/I_Heat_0.dat'
10  'I_Files/I_Bfield_0.dat'
11  'I_Files/I_Accretion_0.dat'
-:--  Cool_Try.in   26% (6,0)    (Fundamental)
```

9: controls which heating mechanisms are used.
10: controls magnetic field evolution: not implemented anymore !
11: controls the accretion rate.

Notice that these last three files have names as I_*_**0**.dat:
this indicates that these files define
- ***no*** heating,
- ***no*** magnetic field evolution (in this case there is no other option !) and
- ***no*** accretion.

The input file line by line (3)

12: contains some parameter which control the output: see next slide.
[*In this file are the parameters, among others, which determine whether the next three files are used or not.*]

Cool_Try.in

```
 OUTPUT FILES:
12 'Model_1/I.dat'
13 'Model_1/Teff_Try.dat'
14 'Model_1/Temp_Try.dat'
15 'Model_1/Star_Try.dat'

-:** Cool_Try.in   Bot (19,0)   (Fundamental)
```

13: output file: contains a few line of general info about the model and then one line per time step:

Teff_Try.dat

Step	Time [years]	Teff at inf [K]	L_phot [erg/sec]	L_nu [erg/sec]	L_heat [erg/sec]
1	1.000E-12	1.135E+07	2.425E+37	4.382E+48	0.000E+00
2	2.500E-12	1.135E+07	2.425E+37	4.382E+48	0.000E+00
3	4.750E-12	1.135E+07	2.425E+37	4.382E+48	0.000E+00
•	•••	•••	•••	•••	•••

etc

This is the file to use to plot a cooling curve

14: output file: contains the full *T* & *L* (and other variables) profiles at determined times [the times at which print out is done are given in the file "`I.dat`" **12**]

15: output file: contains the full profile of the star of time independent variables as r, ρ, P, T_c's, ...

l.dat

Output control: the "I.dat" file

1: already described previously.
2: DEBUG > 0 will print out zillions of screenfull of info about what NSCool is doing. Some hints about it are at the top of the file NSCool.f
3: time step at which the debug print out will begins.
4: PTEFF =1: the output file "Teff_*.dat" is used.
5: PTEMP =1: the output file "Temp_*.dat" is used.
If PTEMP =2: a HUGE output file "Temp_*.dat" is generated. [More on it somewhere else]
6: PSTAR =1: the output file "Star_*.dat" is used.
7, 8, and 9: some integers which can be used to print out more things in "Teff_*.dat".
10: the calculation stops when T_e^∞ drops below this value (in K).
11: initial temperature (in K) for the initial T profile.

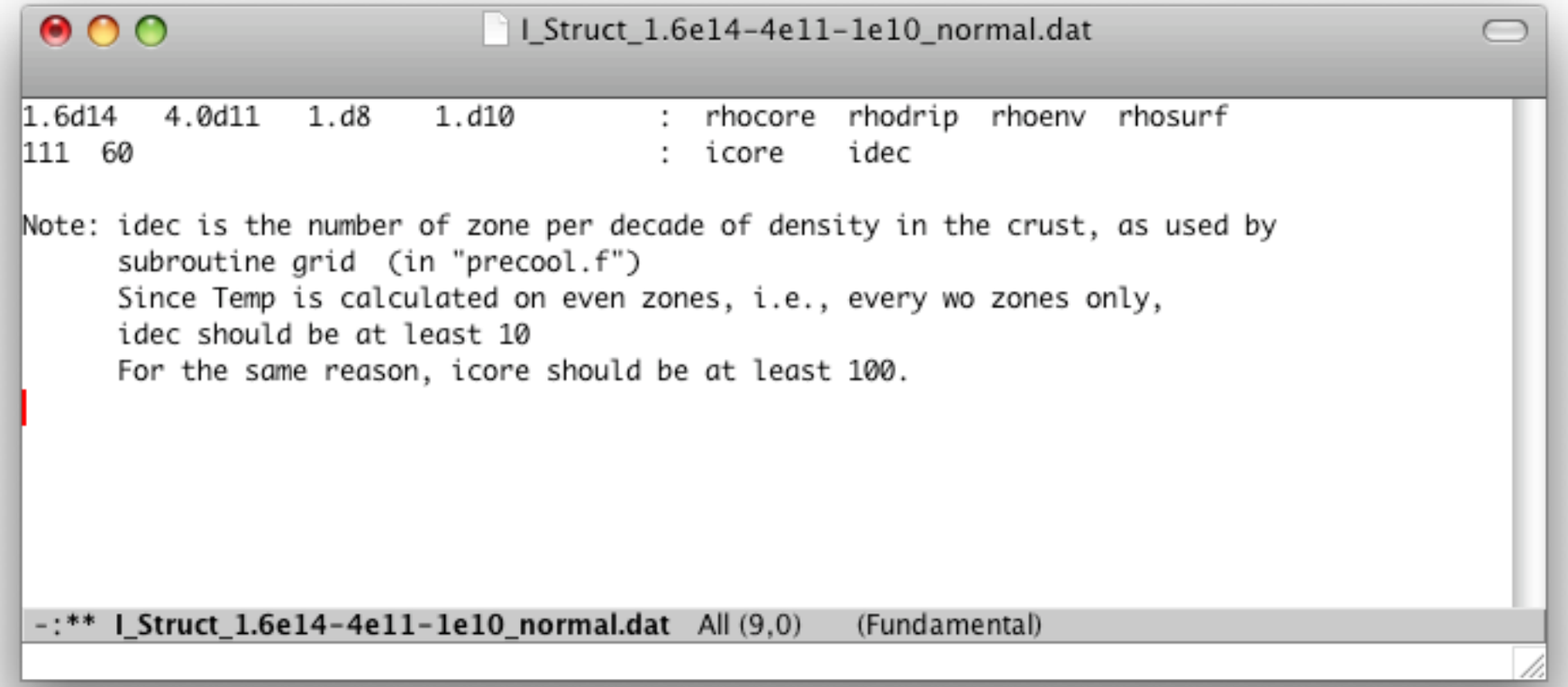
Not used anymore
[but kept here for backward compatibility]

List of times, in years, at which the full profiles will be printed out in the file "Temp_*.dat".

I.dat

```
PRINT OUT CONTROL:-----------------------------------
1       PSCREEN
0.      DEBUG
0       ISTEP DEBUG
1       PTEFF
0       PTEMP
1       PSTAR
1       IDUMP1
111     IDUMP2
421     IDUMP3
1.e4    TEMPMIN
1.e10   TEMPINI
ELECTRON SPECIFIC HEAT in OUTER CRUST CONTROL:------------
0       ICVEL_NODEG
EFFECTIVE MASS CONTROL: for old EOS, not used anymore ----
5       EMNCO
5       EMNCR
3       EMP
INITIAL ROTATIONAL PERIOD:--------------------------------
0.1     p0: initial spin period
TPRINT: times at which you select the profiles -----------
1.0e-10
1.0e+0
3.0e+0
1.0e+1
1.0e+2
1.0e+3
1.0e+4
1.0e+5
1.0e+6
1.0e+7
-:** I.dat       All (32,0)    (Fundamental)
```

l_Struct_*.dat



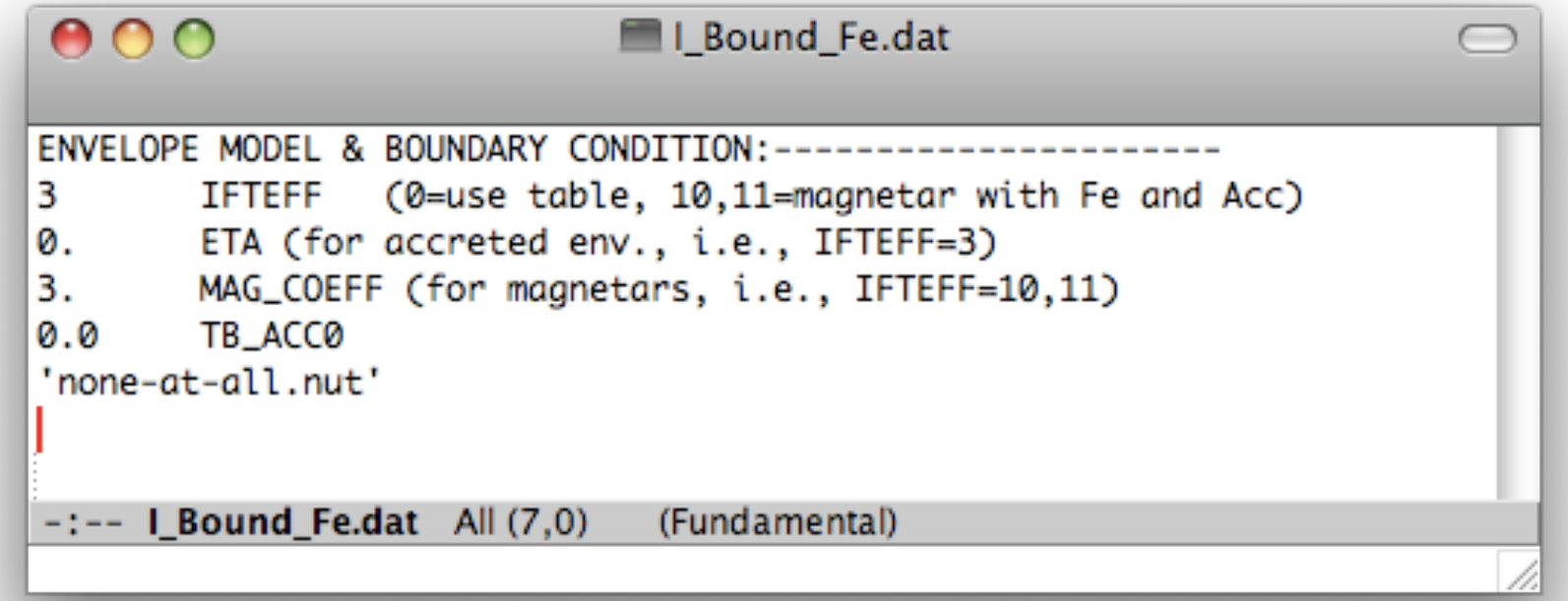
- These parameters are used by subroutine grid (in file precool.f) to redefine the grid.
- Remember that $\mathcal{T}_i$ is defined at i = 1, 3, 5, ..., imax and

$\mathcal{L}_i$ is defined at i = 0, 2, 4, ...,imax-1

- rhocore, rhodrip and rhosurf have obvious meaning and will define density at the grid points icore, idrip and imax.
- rhoenv (at grid point ienv) allows to extend the calculations to low density: the zone at densities between rhoenv and rhosurf will have their density adjusted during the cooling. If rhoenv < rhosurf then nothing happens (as in the sample file above).
- icore defines how many zones there will be in the core, located at radii which will give approximately the same volume in each zone.
- idec defines how many zones per decade in density there will be in the crust.

[icore should be an odd number because $\mathcal{T}_{\rm icore}$ must be defined, and idec should be an even number because $\mathcal{T}_{\rm idrip}$, $\mathcal{T}_{\rm ienv}$, and $\mathcal{T}_{\rm imax}$ must be defined: however the code checks it and adjust icore/ienv]

I_Bound_*.dat



These parameters are used by the function fteff (in the file boundary.f) and define the outer boundary condition, i.e., the "T_e-T_b" relationship. So:
$T_e = T_e(T_b, ..)$ is fteff(Tb, ...)
and both T_e and T_b are local temperatures, not red-shifted ones.

Choices of envelope models (1)

The envelope model being use is defined by the parameter:

IFTEFF =

0 : uses a table, defined by the character string 'none-at-all.nut' (which should be replaced by the exact path and name of the table to use !). See the subroutine fteff_table for the format of the table.

1: uses the 'Te-Tb' relationship of:

Structure of neutron star envelopes
Gudmundsson, E. H.; Pethick, C. J.; Epstein, R. I.
1983, ApJ 272, p. 286

These are envelope models with iron.

2: uses the 'Te-Tb' relationship of:

Cooling of neutron stars - Effects of the finite time scale of thermal conduction
Nomoto, K.; Tsuruta, S.
1987, ApJ 312, p. 711

These are envelope models with iron.

These two models are old, it is better to use the next one (IFTEFF=3);

Choices of envelope models (2)

IFTEFF =

3: uses the 'Te-Tb' relationship of:

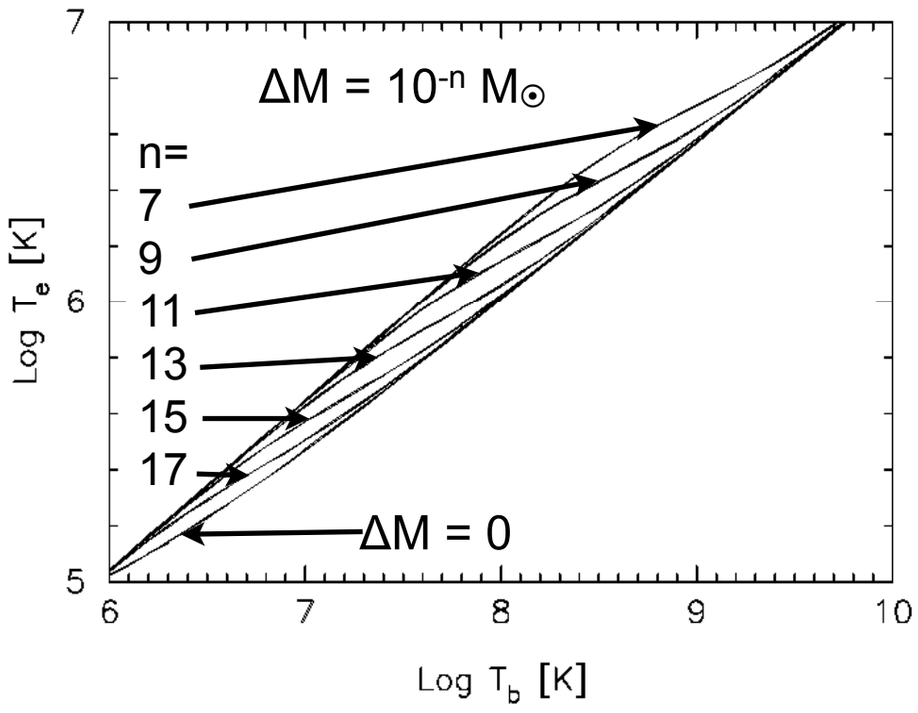
Internal temperatures and cooling of neutron stars with accreted envelopes
Potekhin, A. Y.; Chabrier, G.; Yakovlev, D. G.
1997, A&A 323, p.415

These are envelope models with light elements (and iron+heavier ones at high density).
The amount of light elements is determined by the parameter η (= `ETA` in the file):

$$\eta = g_{s14}^2 \, \Delta M/M = P_{light} / 1.193\times10^{34} \text{ dyne cm}^{-2}$$

where P_{light} is the pressure at the bottom of the light element layer.

Notice that light elements cannot be present at too high densities (e.g., C will burn by pycnonuclear reactions at about 10^{10} g cm^{-3}). So there is a maximum value that η can reach. The formula that Potekhin et al. give saturates when η grows: thus a value as η=1 will give the maximum possible effect of a light element envelope (even if it is physically a wildly unrealistic high value).



Choices of envelope models (3)

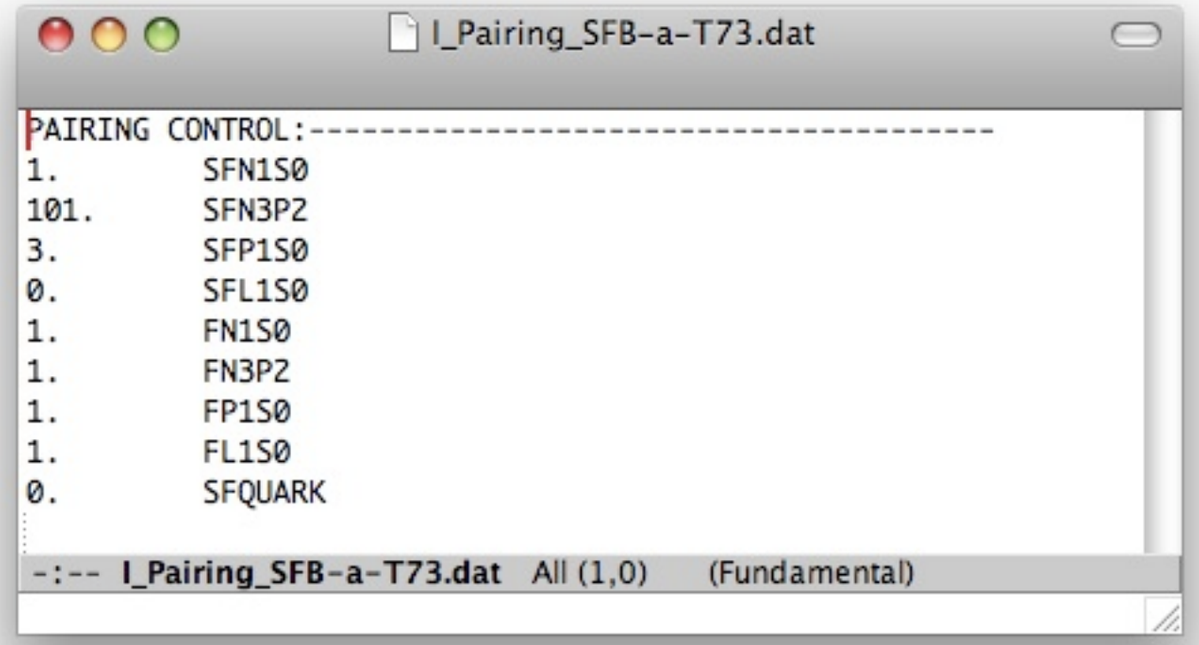
IFTEFF =

4 : more on it later !

5: more on it later !

15: in this case the value of T_b is held constant: this can be used to simulate accreting neutron stars where the outer temperature is controlled by the accretion. The value of T_b is then given by the parameter `TB_ACC0`.

L_Pairing_*.dat

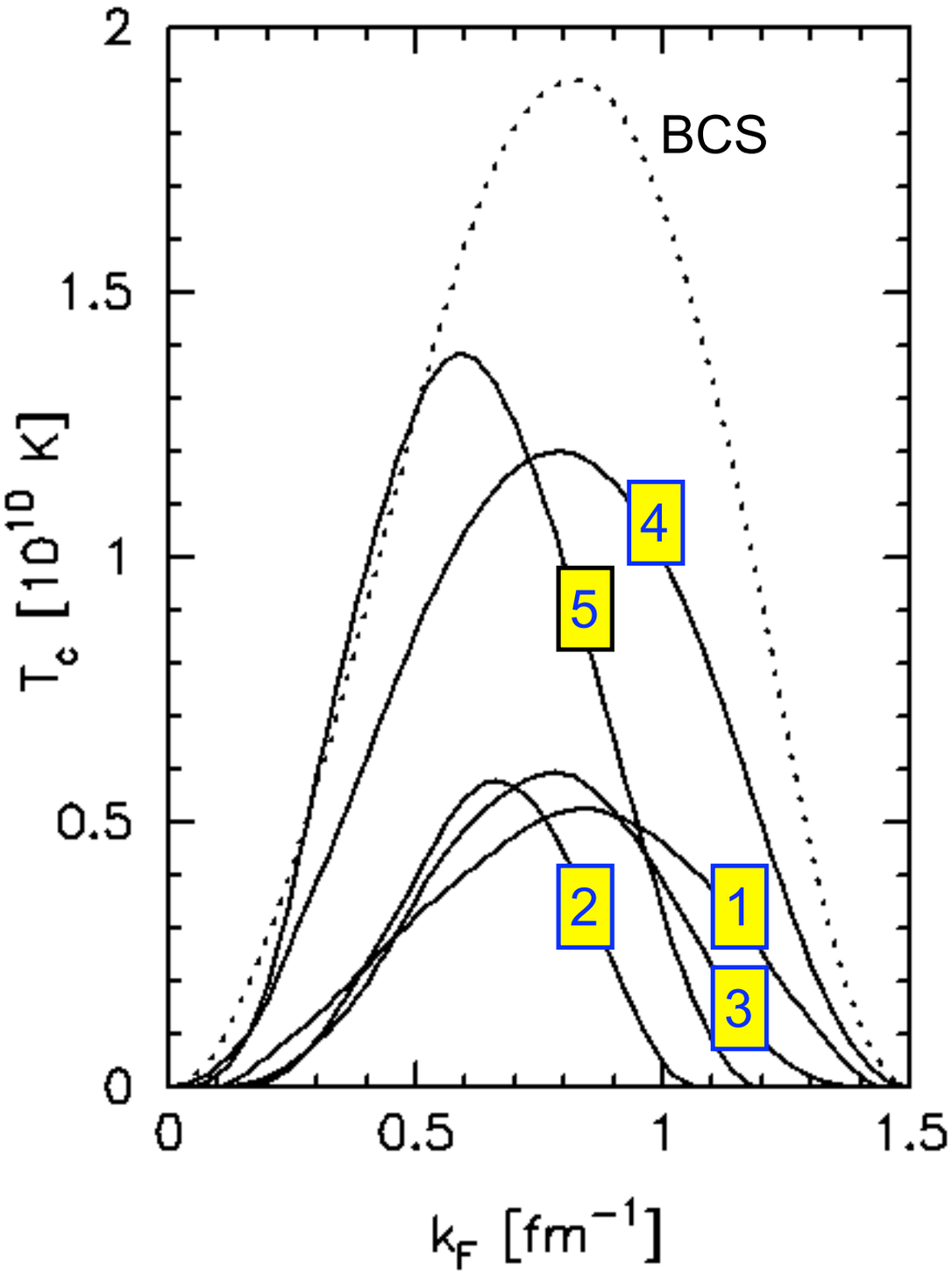


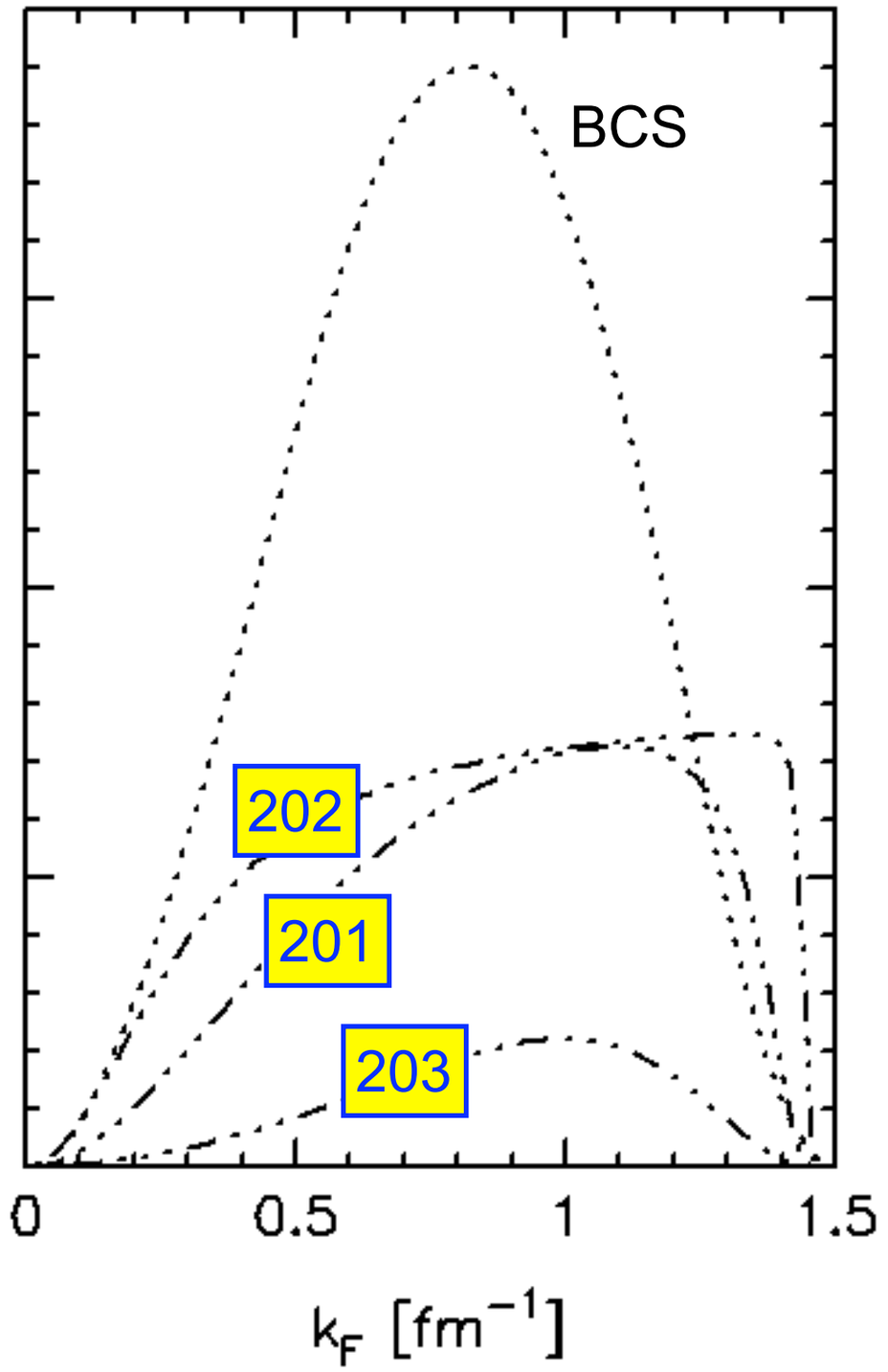
Parameter controls:

`SFN1S0`	neutron $^1S_0$ gap model to be used
`SFN3P2`	neutron $^3P_2$ gap model to be used
`SFP1S0`	proton $^1S_0$ gap model to be used
`SFL1S0`	lambda hyperon (Λ) $^1S_0$ gap model to be used
`FN1S0`	scaling factor for neutron $^1S_0$ gap
`FN3P2`	scaling factor for neutron $^3P_2$ gap
`FP1S0`	scaling factor for proton $^1S_0$ gap
`FL1S0`	scaling factor for lambda hyperon (Λ) $^1S_0$ gap
`SFQUARK`	quark gap (not safely implemented)

[The scaling factors allow to easily play with the sizes of the gaps: $T_c \rightarrow$ scaling factor $\times T_c$]

Neutron $^1S_0$ Gaps





Values of the parameter `SFN1S0`:

	Gap:
1	SFB
2	CCDK
3	WAP
4	GC
5	GIPSF

201	1ns
202	2ns
203	3ns

(See next slide for references)

Neutron $^1S_0$ Gaps

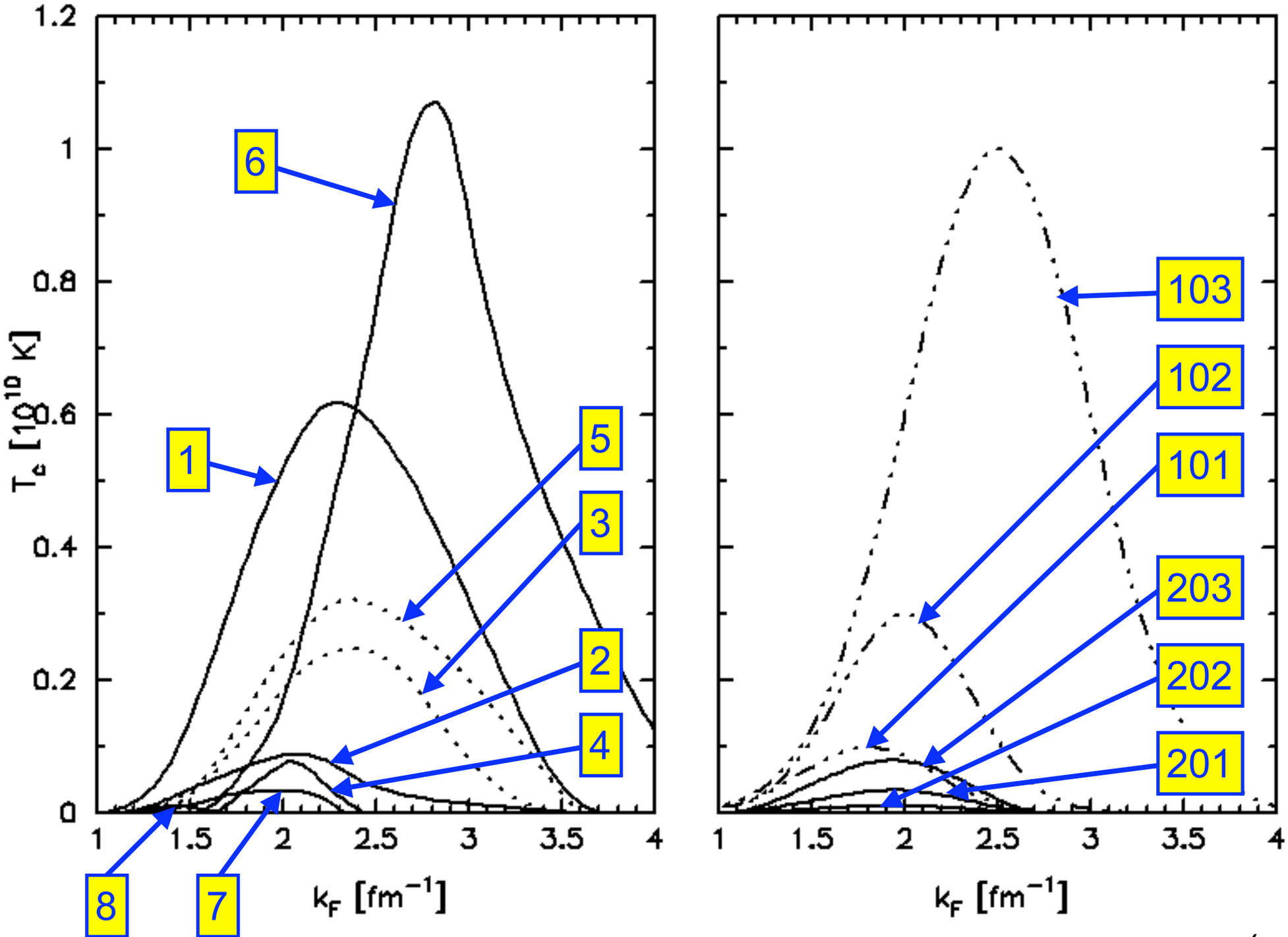
The neutron $^1S_0$ gap has been the most extensively studied to date. The curve “BCS” shows the prediction for the simplest model (in the “BCS approximation”).
Models “SFB”, “CCDK”, & “WAP” go beyond this by including medium polarizations while “GC” & “GIPSF” use quantum Monte-Carlo techniques.

References:

- **SFB**: Schwenk, Friman, & Brown, Nucl. Phys. A713 (2003), p. 191.
- **CCDK**: Chen, Clark, Dave, & Khodel, Nucl. Phys. A555 (1993), p. 59.
- **WAP**: Wambach, Ainsworth, & Pines, Nucl. Phys. A555 (1993), p. 128.
- **GC**: Gezerlis, & Carlson, Phys. Rev. C77 (2008), p. 2801.
- **GIPSF**: Gondolfi, Iliaronov, Pederiva, Schmidt, & Fantoni, Phys. Rev. Lett. 111 (2008), p. 132501.

- **1ns, 2ns 3ns**: *ad hoc* models proposed by
  - Yakovlev, Kaminker, & Gnedin, A&A 379 (2001), p. L5
  - Kaminker, Yakovlev, & Gnedin, A&A 383 (2002), p. 1076
  - Yakovlev, Kaminker, Haensel, & Gnedin, A&A 398 (2002), p. L24.

Neutron $^3P_2$ Gaps



Values of the parameter `SFN3P2`:

Gap:

1	HGRR
2	AO
3	AO (m*=1)
4	T72
5	T72 (m*=1)
6	BCLL
7	EEHJO n.r.
8	EEHJO r.

101	“a”
102	“b”
103	“c”
201	1nt
202	2nt
203	3nt

(See next slide for references)

Neutron $^3P_2$ Gaps

The neutron $^3P_2$ gap is the most important one, but it's actual value is essentially unknown (as the many curve in the previous slide easily convince you)

References:

- **SFB**: Schwenk, Friman, & Brown, Nucl. Phys. A713 (2003), p. 191.
- **CCDK**: Chen, Clark, Dave, & Khodel, Nucl. Phys. A555 (1993), p. 59.
- **WAP**: Wambach, Ainsworth, & Pines, Nucl. Phys. A555 (1993), p. 128.
- **GC**: Gezerlis, & Carlson, Phys. Rev. C77 (2008), p. 2801.
- **GIPSF**: Gondolfi, Iliaronov, Pederiva, Schmidt, & Fantoni, Phys. Rev. Lett. 111 (2008), p. 132501.

- **1ns, 2ns 3ns**: *ad hoc* models proposed by
  - Yakovlev, Kaminker, & Gnedin, A&A 379 (2001), p. L5
  - Kaminker, Yakovlev, & Gnedin, A&A 383 (2002), p. 1076
  - Yakovlev, Kaminker, Haensel, & Gnedin, A&A 398 (2002), p. L24.

l_Neutrino_*.dat

I_Conduct_*.dat

I_Conduct_22-0.1.dat

```
HEAT TRANSPORT CONTROL:-----------------------------------
0       IOPACITY
2       ICON_CRUST
2       ICON_CORE
0       ICONNOTHING2        ! Not used so far
0       ICONNOTHING3        ! Not used so far
0       ICONNOTHING4        ! Not used so far
0       ICONNOTHING5        ! Not used so far
0.0     CONNOTHING1         ! Reserved for strange star
0.0     CONNOTHING2         ! Reserved for strange star
0.1     Q_imp
0.0     CONNOTHING4         ! Not used so far
0.0     CONNOTHING5         ! Not used so far

-:--  I_Conduct_22-0.1.dat   All (1,0)     (Fundamental)
```

[Note: the "_22-0.1" in the file name means that **ICON_CRUST=2**, **ICON_CORE=2** and **Q_imp=0.1**]

IOPACITY :

0: no photon opacity **1**: with photon opacity included in the thermal conductivity

Q_imp: "impurity parameter for e-impurity scattering.

ICON_CORE :

1: simple Flowers & Itoh formula: $$\lambda = 10^{23} \cdot \left(\frac{k_{F,n}}{1.6\,\mathrm{fm}^{-1}}\right) \cdot \left(\frac{T}{10^8\,\mathrm{K}}\right)$$

2: use the full calculation of Yakovlev et al.

[ICON_CRUST: see next slide]

Crust Conductivity

ICON_CRUST :
This will essentially distinguish between the *Itoh et al.* and *Yakovlev et al.* calculations, both in the liquid and the crystal phases:

$\Gamma > \Gamma_c$:
- **1**: e-phonon from *Itoh et al.* + e-impurity from *Yakovlev & Urpin*.
- **2**: e-phonon from *Baiko & Yakovlev* + e-impurity from *Yakovlev & Urpin*.
- **3**: e-phonon from Gnedin et al. (2001: appendix) + e-impurity from *Yakovlev & Urpin*.

$\Gamma < \Gamma_l$:
- **1**: e-ion from *Itoh et al*.
- **2**: e-ion from *Itoh et al*.
- **3**: e-ion from Gnedin et al. (2001: appendix).

$\Gamma_l < \Gamma < \Gamma_c$:
interpolate between the two previous cases (to avoid a discontinuity in λ in cases 1 & 2).
[If you set $\Gamma_l = \Gamma_c$ then, of course, there will be no interpolation !]

If $\rho < 10^7$ g cm^{-3} ("envelope"): none of the above, just use Potekhin et al. (1999).

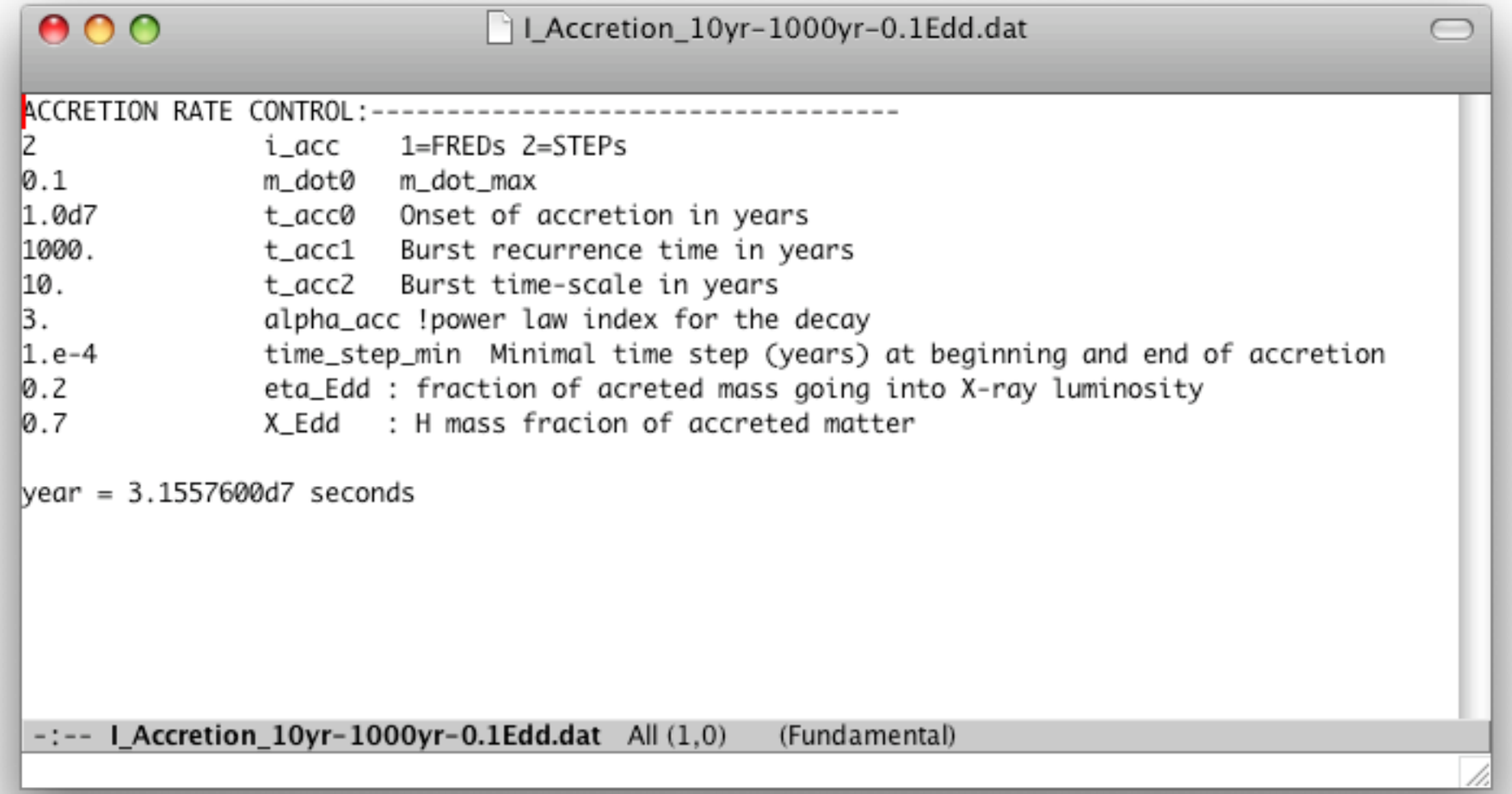
After all this the e-e scattering contribution (Shternin & Yakovlev 2006) is added.

Γ_c = gammacryst > gammaliq = Γ_l are defined in the included file gamma_limits.inc.f
[If you changes these values you MUST recompile all the fortran files: `rm -f *.o` and then `make NSCool.out`)]

Heat_*.dat

I_Bfield_0.dat

l_Accretion_*.dat



These parameters are used by the accretion subroutines, in file `accretion.f` .
See *NSCool_Guide_Accretion* for details.