

Peter Eggleton's
Binary Evolution Code

CVS version

IO Manual

Marc van der Sluys¹, Evert Glebbeek²

¹Northwestern University, ²McMaster University

<http://www.astro.northwestern.edu/~sluys/Eggleton/>

December 12, 2009

Contents

1	IO files	3
2	init.dat	4
3	init.run	9
4	file.mod	12
5	file.mdl	14
6	file.plt	15
7	Creating a ZAMS model	17
8	Creating a ZAHB model	18
9	Variables in SX and PX	19
10	The independent variables	21
11	The difference equations	22
12	The boundary conditions	23

1 IO files

Input files

init.dat Initialisation file. Contains the options to start a stellar model. (fort.22)

init.run Run control file. Controls the start and stop conditions for different models in a run. One can loop over M_1 , $q \equiv \frac{M_1}{M_2}$ and P_1 . Output from different loops is stored in files with different names or in different directories. The file file.list gives an overview of which model is stored where. (fort.23)

Output files

As an example I chose the filename `file.*` for the model files.

file.out1,2 Main output file, showing what the stars are doing at that moment. These files are usefull as ‘screen output’. (fort.1,2)

file.mod Contains a number of complete stellar structure output blocks. A block from this file can serve as input for a next model. (fort.15)

file.plt1,2 Contains stellar evolution data, one model per line. (fort.31,32)

file.mdl1,2 Contains a number of complete stellar structure models, one mesh point per line. (fort.33,34)

file.last1,2 Contains complete structure of last and pre-last model, when lucky, that can serve as input for a next run. (fort.13,14)

file.list Shows the starting time and path of a run and tables the properties of the different models and the filenames or directories in which they are stored. (fort.40)

file.log Shows how the code was terminated, if terminated properly. (fort.8)

Data files

The files `fort.12` and `fort.16-21` are used for data input (like ZAMS, opacities, etc.)

fort.12 `zahb.mod`: Input structure model for post helium-flash models

fort.16 `zams.mod`: Input structure model for ZAMS models

fort.17 `zams.dat`: Helps to find the proper ZAMS model from `zams.mod`

fort.18 `zams.out`: Helps to find the proper ZAMS model from `zams.mod`

fort.19 `zams.mas`: Helps to find the proper ZAMS model from `zams.mod`

fort.20 `phys.z*`: opacity tables for certain metallicity

fort.21 `1t2ubv.dat`:

Temporary files

fort.50 is used to create file.list

2 init.dat

The `init.dat` file contains the initial model parameters. It consists of 31 lines. Single star and TWIN models need 1 lot of these 31 lines, non-TWIN binary models need 2.

```
1 KH2 KR1 KR2 JCH KTH KX KY KZ
2 KCL KION KAM KOP KCC KNUC KCN
3 KT1 KT2 KT3 KT4 KT5 KSV
4 EPS DEL DH0 CDC(1)...CDC(5)
5 KE1 KE2 KE3 KBC KEV KFN KL JH1 JH2 JH3
6-11 KD(11 - 130) - 6 lines
12-18 KD(131 - 260) - 7 lines
19-21 KSX(1 - 45) - 3 lines
22 KN
23-24 KJN(1 - 40) - 2 lines
25 CT1 CT2 CT3 CT(1)...CT(10)
26 CC CN CO CNE CMG CSI CFE
27 CALP CU COS CPS CRD CXB CGR CEA CET
28 CMT CMS CMI CMR CMJ CML CHL CTF CLT
29 CPA CBR CSU CSD CDF CGW CSO CMB
30
31
```

Line 1: KH2 KTR1 KTR2 JCH KTH KX KY KZ

KH2 The number of mesh-points you want; if this differs from KH the code should interpolate in the given model to produce a new one; but you must also set **JCH** to ≥ 2 to implement this change (199)

KR1 The maximum number of iterations allowed on the first timestep (12)

KR2 The maximum number of iterations allowed on later timesteps (12)

JCH If $JCH > 1$, the REMESH initialises the model in various ways:

JCH = 1 Does nothing.

JCH = 2 Initialises some new variables, for instance the mass,

JCH = 3 = (2) + constructs new mesh spacing by interpolation,

JCH = 4 = (3) + initialises composition to uniformity (for ZAMS).

At least in some cases **JMX** in `INIT.RUN` must be 0 if **JCH** > 1 . [**KTH**(1) – (4), *alias* **KTH** – **KZ**:]

KTH $\epsilon_{\text{th}} = KTH * (T \text{ DS}/Dt)$; so you can ignore $T \text{ DS}/Dt$ if you want (1 or 0)

KX $DX(^1\text{H})/Dt = KX * (\text{burning rate of } ^1\text{H})$; so you can ignore the composition change while keeping the energy production (1 or 0)

KY The same, for ^4He (1 or 0)

KZ The same, for ^{12}C and ^{16}O (1 or 0)

Line 2: KCL KION KAM KOP KCC KNUC KCN

KCL(1) – (7) also known as **KCL**, **KION**, **KAM**, **KOP**, **KCC**, **KNUC**, **KCN**:

KCL Unity includes the Coulomb correction to pressure etc; zero suppresses it. (1)

KION EoS does the ionisation of the first KION elements in the list H, He, C, N, O, Ne, Mg, Si, Fe. No other elements are included. KION = 5 is about optimal. Don't try 9. (5)

KAM Not currently used

KOP If unity, code should use spline interpolation in tables of opacity; if zero, simple bilinear interpolation (1)

KCC Not currently used

KNUC Not currently used

KCN If 0, gives standard nuclear network. If 1, gives a CNO-equilibrium fudge for ZAMS models: see FUNCS1 (0)

Line 3: KT1 KT2 KT3 KT4 KT5 KSV

KT(1) – (4) also known as KT1, KT2, KT3, KT4:

KT1 Print internal details of every KT1'th model to screen and `file.mdl` (20 or 200)

KT2 Print internal details at every KT2'th meshpoint of the KT1'th model (screen only) (1 or 2)

KT3 Print KT3 'pages' of details for every KT1'th model to screen (1, 2 or 3)

KT4 Print a five-line summary of every KT4'th model to screen and save every KT4'th evolution model to `file.plt*` (1, 2 or 4)

KT5 Print a one-line summary of each iteration of each model to screen, except for the first KT5 iterations of each model (0 or 2)

KSV an output model is stored in `file.mod (fort.15)` after every KSV'th timestep in a run, in the form needed as input for a further run. The last model of a run is automatically also stored, in `file.last* (fort.13,14)`. (5000)

Line 4: EPS DEL DH0 CDC(1) ... CDC(5)

EP(1) – (3) also known as EPS, DEL, DH0:

EPS The accuracy to which SOLVER is expected to solve the equations (10^{-6})

DEL See Writeup, section 1.5 (10^{-2})

DH0 See Writeup, section 1.5 (10^{-7})

CDC(1) – (5): CDD is The mean increment, r.m.s.-wise, that you would like in one timestep. Different evolutionary phases have different CDD's:

CDC(1): CDD = CDC(1), between ZAMS and helium ignition. (0.01)

CDC(2): CDD = CDC(1)*CDC(2), for evolution during core He burning. (0.25)

CDC(3): CDD = CDC(1)*CDC(3), for further evolution until the He shell nearly catches up with the H shell. (1.0)

CDC(4): CDD = CDC(1)*CDC(4), for double-shell-burning. (4.0)

CDC(5): Unused

Line 5: KE1 KE2 KE3 KBC KEV KFN KL JH1 JH2 JH3

See also Writeup, section 1.5

KE1, KE2 The number of first and second order difference equations respectively

KE3 Subset of KE1 that involves 3 rather than 2 adjacent mesh points (not yet used, keep 0)

KBC The number of boundary conditions

KEV The number of eigenvalues

KFN The number of 'intermediate functions'

KL Determines where to start the calculations: 0 - surface, 1 - centre. Presently, KL=1.

JH1 – JH3 Used for debugging purposes

Line 6-11:

See also Writeup, section 1.5

KD(120) Three lots of 40 integers (2 lines of 20 each), determining which variables, difference equations and boundary conditions are used and in which order:

Line 6-7: Determines which and in which order the [independent variables](#) are used.

Line 8-9: Determines which and in which order the [difference equations](#) are used.

Line 10-11: Determines which in which order the [boundary conditions](#) are used.

Line 12-18:

The same contents as [lines 5-11](#), not currently used. See the end of section 1.5 of Writeup.

Lines 19–21:

KSX(45) The first 15 integers identify the quantities, such as $\log \rho, L, X(^4\text{He}), \dots$, which are to be printed in columns on the first 'page' of structure details for every [KT1](#)'th model. The next two lots of 15 relate to the optional further 'pages'. See section [9](#).

Line 22:

KN The number of variables that will be used for determining the next time step.

Line 23-24:

KJN(1)-KJN(40) The first KN of these identify the variables to be used for determining the next time step, see section [10](#).

Line 25: CT1 CT2 CT3 CT(1) ... CT(10)

CT1 The next timestep cannot (normally) be less than CT1 times present timestep (0.8, 0.9 or 1.0)

CT2 The next timestep cannot be greater than CT2 times present timestep. If both CT1 and CT2 are 1.0, then the timestep is constant, of course (which is useful for constructing a ZAMS by artificial 'mass-gain') – except that if a model fails to converge the timestep will be multiplied by [CT3](#) (1.1, 1.05 or 1.0)

CT3 when the solution package fails to converge, the code retreats to the second-last converged model, and continues with the timestep decreased by the factor CT3. (0.3 or 0.5)

CT(1) – (10) coefficients used in the mesh-spacing function Q :

- CT(1)** Unused (0.00)
- CT(2)** Used for Luminosity weight (0.00)
- CT(3)** Used for Radius weight, together with CT(8) (0.05)
- CT(4)** Used for Pressure weight, together with CT(5,10) (0.05)
- CT(5)** Used for Pressure weight, together with CT(4,10) (0.15)
- CT(6)** Used for Mass weight (0.02)
- CT(7)** Used for Temperature weight, together with CT(10) (0.45)
- CT(8)** Used for Radius weight, together with CT(3) (1.E-4)
- CT(9)** Used for Pressure weight, together with CT(4,5) (1.E15)
- CT(10)** Used for Temperature weight, together with CT(7) (2.E4)

Line 26: CC CN CO CNE CMG CSI CFE

CC, CN, CO, CNE, CMG, CSI, CFE values for initialising $X(^{12}\text{C}), \dots X(^{56}\text{Fe})$, as fractions of the total metallicity Z ($= \text{CZS}$ in `input/phys.z*` (`fort.20`)); only used for ZAMS models, with **JCH** = 4 (0.176, .052, 0.502, 0.092, 0.034, 0.072, 0.072)

Line 27: CALP CU COS CPS CRD CXB CGR CEA CET

CALP The mixing-length ratio (2.0)

CU Along with **COS** and **CPS**, a ‘convective overshoot’ parameter, see **CRD**. (0.1)

COS A convective overshoot parameter for H-burning cores, see **CRD**. Zero implies no overshoot. (0.12)

CPS as **COS**, but for He-burning cores. (0.12)

CRD The diffusion coefficient σ for convective mixing is taken to be **CRD** times the ‘legitimate’ rate from mixing-length theory; except that an approximate multiple of $[\nabla_r - \nabla_a]^{1/3}$ is replaced by the same multiple of $[\nabla_r - \nabla_a + \nabla_{\text{OS}}]^2$, where

$$\nabla_{\text{OS}} \equiv \frac{\text{COS}}{(2.5 + 20\beta_* + 16\beta_*^2)(\text{CU} \cdot \partial \log m / \partial \log P + 1)}, \beta_* \equiv \frac{P_{\text{rad}}}{P_{\text{gas}}}.$$

The usual **CRD** is 10^{-4} .

CXB Defines the boundary of a core to be at $X(^1\text{H})$ or $X(^4\text{He}) = \text{CXB}$; for printout and envelope binding energy (0.15)

CGR Defines the boundary between a convection zone and a semiconvection zone, for printout purposes only, to be at $\nabla_r - \nabla_a + \nabla_{\text{OS}} = \text{CGR}$ (0.001)

CEA A constant energy rate **ENC** (in `init.run`) can be added to $\epsilon_{\text{nuc}} + \epsilon_{\text{th}} - \epsilon_{\nu}$. An increasing **ENC** can push a star back from the ZAMS to the Hayashi track. **CEA** and **CET** determine how **ENC** changes with time (1.0E2)

CET The equation for the growth of **ENC** with time is $d\text{ENC}/dt = \text{ENC} \times \text{CET} \times (1 - \text{ENC}/\text{CEA})$, so that **ENC** increases exponentially on the assigned timescale $1/\text{CET}$ (yr), until saturating at $\text{ENC} \sim \text{CEA}$. (1.0E-6)

Line 28: CMT CMS CMI CMR CMJ CML CHL CTF CLT

\dot{M} At the surface,

$$\dot{M} = - \text{CMT} \cdot \xi - \text{CMS} \cdot [\log(r/r_{\text{lobe}})]^3 + \text{CMI} \cdot m - \text{CMR} \cdot 1.3 \times 10^{-5} \cdot L \cdot m \cdot |E_{\text{BE}}| \\ - \text{CMJ} \cdot \dot{M}_{\text{JNH}} - \text{CML} \cdot \zeta(L, r, m, P_{\text{rot}})$$

($[X] \equiv X$ if $X > 0$ and 0 if $X < 0$)

CMT one of two versions of MT by RLOF. CMS & CMT are *alternatives*; set one to zero (0.0, or 1.0D-2–1.0D2 for stars of increasing mass (?))

CMS one of two versions of MT by RLOF. CMS & CMT are *alternatives*; set one to zero (0.0, or 1.0D4)

CMI a constant mass-gain/loss rate, for running up or down the ZAMS, (yr^{-1}) (0.0, $\pm 5.0\text{D}-9$ or $\pm 1.0\text{D}-6$)

CMR a Reimers-like mass-loss rate, dimensionless: $\dot{M} = \text{CMR} \times M \times \max\left(\frac{1.3 \times 10^{-5} L}{U_{\text{bind}}}, \frac{10}{\tau_{\nu}}\right)$ (0.0 or 0.2–1.0)

CMJ a mass-loss rate for luminous stars, dimensionless (de Jager et al 1988) (0.0 or 1.0)

CML a mass-loss rate as obtained from a simplistic dynamo theory, dimensionless (0.0 or 1.0)

CHL A factor multiplying the rate of ang. mom. loss associated with the rate of mass loss ζ , according to the same dynamo model. (0.0 or 1.0)

CTF A factor multiplying an expression for the rate of tidal friction. (0.0 or 0.01)

CLT A coefficient used in the estimation of heat flux between components in contact. It doesn't really work yet (or does it?).

Line 29: CPA CBR CSU CSD CDF CGW CSO CMB

CPA ‘partial accretion’: the fraction of one star’s wind that is accreted by the other. (0.0)

CBR ‘bipolar re-emission’: the fraction of material accreted by a star that is ejected in bipolar jets. Needed for CVs, LMXBs. (0.0)

CSU ‘spin-up’, specifically of the gainer due to accretion. CSU is the specific angular momentum (AM) relative to orbital (OAM), taken out of the orbit by material leaving the L1 point, acquiring AM due to Coriolis force, and landing on the other star - so OAM is converted to gainer’s internal AM. Does not seem to work properly... yet. (0.0)

CSD ‘spin-down’; the same process also spins down the loser, I suppose, though not by as much. Does not seem to work properly... yet. (0.0)

CDF this is used to convert a step-function into a ‘smoothed’ step function: see Writeup p.27. (0.01)

CGW A switch to turn gravitational radiation on and off (0.0 or 1.0)

CSO A switch to turn spin-orbit coupling on and off (0.0 or 1.0)

CMB A multiplication factor to determine the strength of an alternative magnetic braking law, currently the one by Rappaport, Verbunt & Joss, 1983. (0.0 – 1.0)

Line 30: Unused

Line 31: Unused

3 `init.run`

The `init.run` file contains parameters that control the run. It consists of 8 lines. You have to decide on each of four options, each giving two alternatives. The options are

- a) single stars or binary stars
- b) new, i.e. starting from scratch (ZAMS), or old, e.g. starting from the end of a previous run.
- c) independent evolution, or *simultaneous* evolution, of the components.
- d) a ‘one-shot’ or ‘grid’ run. A grid means several runs, one after the other (but simultaneous using the massively parallel version, not described here), with the three parameters of primary mass, mass ratio and orbital period being cycled through. One-shot means what it says.

Not all 16 possibilities make sense: e.g. if you are doing simultaneous evolution, you won’t want single stars. Many but not all of the remaining possibilities should be viable.

Line 1: **ISB, KTW, IOP1, IM1, IOP2, IM2, KPT, KP**

ISB single or binary. $ISB = 1$ implies only single stars to be computed; $ISB = 2$ gives binaries. For *single* stars, you may still use the outer (first) cycle for masses. The inner 2 cycles are automatically set to do only one case each. The mass ratio and the period are of course virtually ignored, but have to be supplied. The period should be so large that there is no danger of RLOF (e.g. $XL = 7.0$, meaning a period of $\sim 10^7$ d).

KTW 1 for normal operation; 2 for TWIN mode, where both stars are solved *simultaneously*.

IOP1 the number (13 – 16) of the file (fort.13 – fort.16) where the initial model for *1 is to be taken from. ZAMS models are on fort.16.

IM1 the sequential number of the model required on fort.IOP1. This is computed automatically, from later data, if the ZAMS file fort.16 is used, so that if IOP1 is 16, it doesn’t matter what value you give for IM1, but you have to give a value.

IOP2 as IOP1, but for *2.

IM2 as IM1, but for *2.

KPT, KP KPT is the maximum number of timesteps for each component (2000 to 4000 for fairly complete evolution). Do approximately KP of *1, then enough of *2 to catch up with *1, then another \sim KP of *1, etc, so that if *2 breaks down before *1 you don’t waste a lot of calculation on *1. You will seldom get *exactly* the number of timesteps that you ask for. For single stars, KP is set to KPT automatically.

Line 2: **ML1, DML, KML**

Line 3: **QL1, DQL, KQL**

Line 4: **XL1, DXL, KXL**

Lines 2 – 4 are parameters for 3 nested loops (mass, mass ratio, and initial period) to be run through. Each loop has: starting value; increment; number of cases (1 more than the number of increments).

- The first loop is: \log_{10} (mass, solar units), starting at ML1, increasing by steps of DML to $ML1 + (KML - 1) \cdot DML$
- The second loop is: \log_{10} (mass ratio in sense {larger/smaller}), starting at QL1, increasing by steps of DQL to $QL1 + (KQL - 1) \cdot DQL$
- The third loop is: $X \equiv \log_{10}(\text{orbital period} / \text{period necessary for *1 to fill its Roche lobe when still on the ZAMS})$, starting at XL1, increasing by steps of DXL to $XL1 + (KXL - 1) \cdot DXL$.
- Obviously you can arrange for only one model (single or binary) to be done, by taking $KML = KQL = KXL = 1$.

Line 5: ROT, KR and EX

ROT, KR • rotational period for each star = rotational breakup period * 10**ROT, if KR = 1
• rotational period for each star = max(1.05 * rotational breakup period, orbital period * 10**ROT),
if KR = 2.

EX the initial eccentricity.

Line 6: SM, DTY, AGE, PER, BMS, ECC, P1, ENC, JMX

(a.k.a. AX(1-8), JMX). The AX's are optional replacements for the values of SM, ..., ENC that the code would normally pick up in fort.IOP1 from some previous run, or from the ZAMS library (`fort.16`). JMX similarly is an optional replacement for JMOD. They are only applied if they are non-negative. Thus you can replace only one, or several.

SM Primary mass [M_{\odot}]

DTY Time step [yr]

AGE Model age [yr]

PER Orbital period [d]

BMS Total binary mass [M_{\odot}]

ECC Orbital eccentricity

P1 Spin period of the primary [d]

ENC Artificial energy rate, see [CEA](#) and [CET](#) in line 27 of `init.dat` [?]

JMX New model number (JMOD)

Line 7–9:

The last three lines are a set of 21 criteria (UC(1–21)) to determine when the run is to be ended (e.g. when the age is greater than 2×10^{10} yr), or when some special procedure should be initiated (e.g. the He-flash evation). You'll have to read the end of `printb.f` to figure them out completely. In many cases the code is stopped by changing the termination code *JO*.

Line 7:

UC(1): (rlf1) Terminate if FLR(=RLF?) (Sect. 6, nr 29) of star 1 exceeds this number (*JO* = 4) (0.1)

UC(2): (age) Terminate if the age of the model in years exceeds this number (*JO* = 5) (2e10)

UC(3): (LCarb) Terminate if $L_C >$ this number (*JO* = 6) (100)

UC(4): (rlf2) Terminate if FLR(=RLF?) (Sect. 6, nr 29) of star 2 exceeds this number (*JO* = 7) (0.2)

UC(5): (LHe) Initiate He-flash evation if $L_{\text{He}} >$ this number, together with UC(6) (*JO* = 8) (1e3, lower for $M_* \approx 2M_{\odot}$)

UC(6): (rho) Initiate He-flash evation if $\log \rho_c >$ this value (?), together with UC(5) (*JO* = 8) (5.3)

UC(7): (MCO) Terminate if degenerate CO-core exceeds this mass, together with UC(8) (*JO* = 9) (1.2)

Line 8:

UC(8): (rho) Terminate if $\log \rho_c >$ this value (?), together with UC(7) ($JO = 9$) (6.3)

UC(9): (mdot) Terminate if $|\dot{M}| >$ UC(9) * M_*/τ_{KH} ($JO = 10$) (3e12)

UC(10): (XHe) Change eps (next number) if the core Helium abundance drops below this number (0.15)

UC(11): (eps) If $Y_{\text{core}} <$ XHe (previous number), set **EPS** to this number. Do not use, keep this number 1e-6!
(1e-6)

UC(12): (dtmin) Terminate if $\Delta t <$ dtmin (in seconds?) (1e6)

UC(13): (sm8) The total mass the post-He-flash model should get, can also be used manually! (1e3)

UC(14): (vmh8) The He-core mass the post-He-flash model should get, can also be used manually! (1e3)

Line 9:

UC(15): (XH) If > 0 : terminate if the core H-abundance drops below this value, you can e.g. stop at the
TAMS ($JO = 51$) (0.0)

UC(16–21): Unused

4 file.mod

This file contains stellar structure output, that can be used as input. The files file.last1,2 have the same format.

The file consists of one or more blocks, starting with a single line with 13 model properties and followed by a block with one line per mesh point with the independent variables. This block contains 24 columns of which only part is used. Some of them are 'eigenvalues' and have the same value for every mesh point.

First line

The first line of the file contains the 13 numbers:

1. M_1 , the mass of the primary [M_\odot]
2. Δt , time step [yr]
3. t , age of the model [yr]
4. P_{orb} , the orbital period [day]
5. BMS , the total binary mass [M_\odot]
6. e , the orbital eccentricity
7. P_{rot} , the rotational period [day]
8. enc , artificial energy term [?]
9. kh , the number of mesh points and thus rows in the stellar structure block below
10. kp , the total number of models to calculate
11. $jmod$, the current model number
12. jb , the number of this star in the binary [1 or 2]
13. jin , the number of independent variables and thus columns in the stellar structure block below [24 for non-TWIN, 40 for TWIN]
14. jf , do or do not overwrite I and ϕ (see below). Just keep it 0. [0 or 2]

Blocks of stellar structure

Each block contains the contents of the variable **H**: 24 models for non-TWIN models, and 40 for TWIN models. Columns 1–16 are reserved for the primary, 17–25 for binary parameters and 26–40 have the same content as 1–16, but for the secondary in the TWIN case. In the loop over all meshpoints in **printb**, the variable **Q(1-24)** contains the same data as **H(1-24,I)** (or the corresponding variables for the secondary in a TWIN model) for each meshpoint **I**. Each line represents a mesh point, the first one usually the **surface** of the star. The 'eigenvalues' are marked with (*EV*). The columns are:

1. $\ln f$, a dimensionless quantity closely related to electron degeneracy: for the case where electrons are non-degenerate and non-relativistic, $f \sim 10^8 \rho / T^{1.5}$
2. $\ln T$, logarithmic temperature [K]
3. X_{16} , mass abundance fraction of ^{16}O
4. m , mass [10^{33} g]
5. X_1 , mass abundance fraction of ^1H
6. C , the gradient of mesh-spacing function $Q(f, T, m, r)$ with respect to meshpoint number K (*EV*).
7. $\ln r$, logarithmic radius [10^{11} cm]
8. L , luminosity. Not logged, because it may be negative [10^{33} erg s $^{-1}$]
9. X_4 , mass abundance fraction of ^4He
10. X_{12} , mass abundance fraction of ^{12}C
11. X_{20} , mass abundance fraction of ^{20}Ne
12. I , the moment of inertia of the interior material [10^{55} g cm 2]
13. P_{rot} , the rotation period (days) of the star (*EV*).
14. ϕ - the centrifugal-gravitational potential [erg]
15. ϕ_s , the potential at the surface, minus the potential on the L1 surface (*EV*) [erg]
16. X_{14} , mass abundance fraction of ^{14}N
17. H_{orb} , the orbital angular momentum (*EV*) [10^{50} gm cm 2 s $^{-1}$]
18. e , the orbital eccentricity (*EV*)
19. F , the flux of mass towards or away from the other star; a function of depth and zero below L1 [10^{33} g s $^{-1}$]
20. <empty>
21. <empty>
22. <empty>
23. <empty>
24. <empty>
25. – 40.: The same as variables 1–16, but for the secondary in case of a TWIN model, otherwise empty (since *jin* (above) equals 24 in that case).

5 file.mdl

This file contains stellar structure output.

The file starts with one line of 3 numbers, followed by blocks that also start with one line, but with two numbers. The rest of each block are 20 or mostly 21 columns.

First line

The first line of the file contains the three numbers:

1. Number of grid points (each line of the column part of the blocks is one grid point)
2. Number of variables (is the number of columns in the blocks)
3. $D_{\text{overshoot}}$ (?)

Blocks of stellar structure

Each block starts with one line with two values:

1. Model number
2. t , model age [yr]

The first line of each block is followed by 21 (sometimes 20, depending on the second number of the first line of the file. In that case, the last variable, U_{int} , is missing) columns with structure parameters, one line for each grid point. The total number of grid points is given by the first number of the first line of the file. The columns are:

1. M , mass coordinate [M_{\odot}]
2. R , radius coordinate [R_{\odot}]
3. P , pressure [dyn cm^{-2}]
4. ρ , density [g cm^{-3}]
5. T , temperature [K]
6. κ , opacity [$\text{cm}^2 \text{g}^{-1}$]
7. $\nabla_{\text{ad}} = \left(\frac{\partial \log T}{\partial \log P} \right)_{\text{ad}}$, adiabatic temperature gradient [-]
8. $\nabla_{\text{rad}} - \nabla_{\text{ad}}$, temperature gradient difference [-]
9. — 15. Abundances of 9: H, 10: He, 11: C, 12: N, 13: O, 14: Ne, 15: Mg
16. L , total luminosity [L_{\odot}]
17. ε_{th} , energy generation by contraction [$\text{erg g}^{-1} \text{s}^{-1}$]
18. ε_{nuc} , energy generation by nuclear reactions [$\text{erg g}^{-1} \text{s}^{-1}$]
19. ε_{ν} , energy generation in neutrinos [$\text{erg g}^{-1} \text{s}^{-1}$]
20. S , entropy [cgs]
21. U_{int} , internal energy [erg (/gm?)] or L/L_{edd} ?

6 file.plt

This file contains stellar evolutionary properties, for one structure model per line. The first line contains the number of columns in the output block. The block currently contains 81 columns, with the following contents:

1. JMAD, Model number
2. t , Age [yr]
3. Δt , time step [yr]
4. M , stellar mass [M_{\odot}]
5. M_{He} , helium core mass [M_{\odot}]
6. M_{CO} , carbon-oxygen core mass [M_{\odot}]
7. M_{ONe} , oxygen-neon core mass [M_{\odot}]
8. $\log R$, stellar radius [R_{\odot}]
9. $\log L$, stellar luminosity [L_{\odot}]
10. $\log(T_{\text{eff}})$, effective temperature [K]
11. $\log T_{\text{c}}$, central temperature [K]
12. $\log T_{\text{max}}$, maximum temperature [K]
13. $\log \rho_{\text{c}}$, central density [g cm^{-3}]
14. $\log \rho_{T_{\text{max}}}$, density at $T=T_{\text{max}}$ [g cm^{-3}]
15. U_{bind} , binding energy of H envelope [erg/M_{\odot}]
16. L_{H} , luminosity by hydrogen burning [L_{\odot}]
17. L_{He} , luminosity by helium burning [L_{\odot}]
18. L_{C} , luminosity by carbon burning [L_{\odot}]
19. L_{ν} , neutrino luminosity [L_{\odot}]
20. L_{th} , luminosity by release of thermal energy [L_{\odot}]
21. P_{rot} , rotational period [days]
22. VK2, $K^2 \equiv \frac{I}{MR^2}$, with I the moment of inertia
23. R_{cz} , Depth (?) of convective envelope [R_{*}]
24. dR_{cz} , Thickness (?) of convective envelope [R_{*}]
25. TET, Convective turnover timescale
26. RAF, Alfven radius
27. BP, poloidal magnetic field
28. P_{orb} , orbital period [days]
29. FLR = $\log(r_{*}/r_{\text{rl}})$, relative Roche Lobe Radius
30. F1, $\sim \Phi_{\text{surf}} - \Phi_{\text{L1}}$ [erg] ?

31. \dot{M} , total mass loss [$M_{\odot} \text{ yr}^{-1}$]
32. \dot{M}_{wind} , wind mass loss [$M_{\odot} \text{ yr}^{-1}$]
33. \dot{M}_{mt} , mass transfer rate [$M_{\odot} \text{ yr}^{-1}$]
34. H_{orb} , orbital angular momentum [$10^{50} \text{ g cm}^2 \text{ s}^{-1}$]
35. dH_{orb}/dt , total orbital angular momentum loss rate [$10^{50} \text{ g cm}^2 \text{ s}^{-2}$]
36. dH_{gw}/dt , change in H_{orb} due to gravitational waves [$10^{50} \text{ g cm}^2 \text{ s}^{-2}$]
37. dH_{wi}/dt , change in H_{orb} due to wind mass loss [$10^{50} \text{ g cm}^2 \text{ s}^{-2}$]
38. dH_{so}/dt , change in H_{orb} due to spin-orbit coupling [$10^{50} \text{ g cm}^2 \text{ s}^{-2}$]
39. dH_{ml}/dt , change in H_{spin} due to non-conservative mass transfer [$10^{50} \text{ g cm}^2 \text{ s}^{-2}$]
40. M_{comp} , companion mass [M_{\odot}]
41. e , orbital ellipticity
42. — 48. Surface abundances of: 42:H, 43:He, 44:C, 45:N, 46:O, 47:Ne, 48:Mg
49. — 55. T_{max} abundances of: 49:H, 50:He, 51:C, 52:N, 53:O, 54:Ne, 55:Mg
56. — 62. Central abundances of: 56:H, 57:He, 58:C, 59:N, 60:O, 61:Ne, 62:Mg
63. — 68. Convection zone boundaries; > 0 : beginning, < 0 end of zone (max. 3 sets)
69. — 74. Semi-convection zone boundaries; > 0 : beginning, < 0 end of zone (max. 3 sets)
75. — 80. Nuclear energy production zone ($\varepsilon_{\text{nuc}} > \varepsilon_{\text{tresh}}$) boundaries; > 0 : beginning, < 0 end of zone (max. 3 sets)
81. Q_{conv} , the mass fraction of the convective envelope
82. T_{c} , central temperature (K) ?
83. STRMDL: a structure model is stored (1.0) or not (0.0)
84. Binding energy due to gravitational energy (erg/ M_{\odot} ?)
85. Binding energy due to Internal energy (erg/ M_{\odot} ?)
86. Binding energy due to Recombination energy (erg/ M_{\odot} ?)
87. Binding energy due to H_2 dissociation energy (erg/ M_{\odot} ?)
88. Entropy in core (cgs)
89. Entropy at $T=1e5K$ (cgs)

7 Creating a ZAMS model

In order to create a (ZAMS) model of certain mass, or to obtain a series of ZAMS models, one can use the RMG mass loss/gain parameter in the `init.dat` file. This parameter gives a mass loss or mass gain that is proportional to the mass of the star.

The method is as follows:

- Choose an existing input model with a mass close to the desired mass
- Set the CMI parameter in `init.dat` to the desired value (usually $\pm 5 \times 10^{-9}$)
- Make sure the time step doesn't change (`CT1 = CT2 = 1.00` in `init.dat`)
- Calculate the factor f with which you want to change the mass to get from the model you have to the model you want. (If you have $1.00 M_{\odot}$ and want $1.02 M_{\odot}$, $f = 1.02$)
- Calculate the approximate number of steps you need to take for a time step size $dt_0 = 10^3$ yr and the CMI above: $N_0 = \frac{\ln f}{\text{CMI } dt_0}$
- Choose a (nice, round, but at least) integer number of steps $N \approx N_0$
- Calculate the true time step for N steps: $dt = \frac{\ln f}{\text{CMI } N}$
- Fill in the values for dt and N in `init.run`
- Run the model for N steps and check the final mass in `file.mod`

$$dt = \frac{\ln\left(\frac{M_f}{M_i}\right)}{N \text{ CMI}}$$

The fortran program `makezams.f` (see [website](#)) is supposed to do all the above.

If all goes well, you'll end up with the mass slightly off. You can give your model the exact mass you want by switching off the wind, put the desired mass in `init.run` and run another 10 models or so.

If the change in mass is less than expected, you may have chosen your timestep too long, so that the code does not converge, recalculates the model with a smaller timestep and continues with this smaller timestep (since it is not allowed to change).

8 Creating a ZAHB model

In order to create a ZAHB model, for instance because the format of the input files has changed or because we want a different metallicity, we can use the following recipe. Most of our work is actually already done by test job 7.

- Evolve a $2.25M_{\odot}$ star until it starts core helium burning. Do not allow the helium to be consumed (KY=0).
- Start mass loss until the star is down to about $0.4M_{\odot}$. These two tasks are covered by test job 7.
- Put the starting model and an appropriate init.dat file in input/zahbZ.mod, where Z is the metallicity (02 = 0.02, etc.)
- Test the result for a $1M_{\odot}$ model (test job 3).

9 Variables in SX and PX

These quantities are calculated in `printb` and stored in the variables `SX` and `PX`. In the loop over all meshpoints, `SX(J, IKK)` is the previous value of `PX(J)`, from the previous meshpoint. `IKK` runs from 1 to `NM`, the number of meshpoints, or from the centre to the surface of the star. (In the same loop, the variable `Q(1-24)` contains the values of `H(1-24,I)` for meshpoint `I`, see Sect. 4).

1. ψ : degeneracy parameter ?
2. P: Pressure
3. ρ : Mass density
4. T: Temperature
5. κ : Opacity
6. ∇_{ad} : Adiabatic temperature gradient $\left(\frac{\partial \log T}{\partial \log P}\right)_{\text{ad}}$
7. ∇ : True temperature gradient $\frac{d \log T}{d \log P}$
8. $\nabla_{\text{rad}} - \nabla_{\text{ad}}$: Difference between the radiative and adiabatic ∇ 's
9. M: Mass
10. H^1 : Hydrogen abundance
11. He^4 : Helium abundance
12. C^{12} : Carbon abundance
13. N^{14} : Nitrogen abundance
14. O^{16} : Oxygen abundance
15. Ne^{20} : Neon abundance
16. Mg^{24} : Magnesium abundance
17. R: Radius
18. L: Luminosity
19. E_{th} : Thermal energy generation rate
20. E_{nuc} : Nuclear energy generation rate
21. E_{ν} : Energy loss rate in neutrinos
22. dM : Shell mass
23. Not used
24. $\frac{n}{(n+1)} = \frac{d \log \rho}{d \log P}$: Homology invar.
25. $U_{\text{hom}} = \frac{d \log R}{d \log P}$: Homology invar.
26. $V_{\text{hom}} = \frac{d \log M}{d \log P}$: Homology invar.
27. U: Internal energy
28. S: Entropy

29. L/L_{edd} : Luminosity relative to Eddington
30. $w_{\text{conv}} \times l$, w_{conv} : convective velocity, l : mixing length
31. μ : Mean molecular weight
32. wt: ?
33. ν_e : $\frac{1}{\mu}$: per free electron
34. $\nu_{e,0}$: $\frac{1}{\mu_0}$: per all electrons
35. w_{conv} : convective velocity
36. M.I.: Moment of Inertia
37. ϕ : centrifugal-gravitational potential
38. F_m : Mass flux towards or away from the other star
39. DGOS: $\nabla_r - \nabla_a + \nabla_{\text{OS}}$ modified schwarzschild criterion: if > 0 : convection (but Writeup says: not used...)
40. DLRK: Heat transfer due to differential rotation
41. $\Delta(\text{enth})$: Difference in enthalpy between star 1 and 2
42. XIK
43. V^2 : $\Delta\Phi_{\text{surf}}$ between star 1 and 2
44. FAC2 $\sim V^2$?
45. FAC1 $\sim V^2$?
46. ...
47. ...
48. ...
49. ...
50. RPP: Reaction rate: pp chain effective $2p \rightarrow 1/2 \text{ He4}$
51. RPC: Reaction rate: effective $\text{C12} + 2 p \rightarrow \text{N14}$
52. RPNG: Reaction rate: effective $\text{N14} + 2p \rightarrow \text{O16}$
53. RPN: Reaction rate: effective $\text{N14} + 2p \rightarrow \text{C12} + \text{He4}$
54. RPO: Reaction rate: effective $\text{O16} + 2p \rightarrow \text{N14} + \text{He4}$
55. RAN: Reaction rate: effective $\text{N14} + 3/2 \text{ He4} \rightarrow \text{Ne20}$
56. ...
57. ...
58. ...
59. ...
60. N^2 : Brunt-Väisälä frequency squared

10 The independent variables

- (1) $\ln f$ - a dimensionless quantity closely related to electron degeneracy: for the case where electrons are non-degenerate and non-relativistic, $f \sim 10^8 \rho / T^{1.5}$
- (2) $\ln T$ - logarithmic temperature (Kelvins)
- (3) X_{16} - fractional abundance by mass of ^{16}O
- (4) m - mass (10^{33} gm)
- (5) X_1 - the abundance of ^1H
- (6) C - the gradient of mesh-spacing function $Q(f, T, m, r)$ with respect to meshpoint number K . C does not vary with K , the meshpoint number, although it varies with time. It is in effect an eigenvalue
- (7) $\ln r$ - logarithmic radius (10^{11} cm)
- (8) L - luminosity (10^{33} erg/s). Not logged, because it may be negative
- (9) X_4 - the abundance of ^4He
- (10) X_{12} - the abundance of ^{12}C
- (11) X_{20} - the abundance of ^{20}Ne

For a more sophisticated binary, including mass loss, magnetic braking, rotation (uniform but time-varying) and tidal friction, a further 7 variables are stored:

- (12) I - the moment of inertia of the interior material (10^{55} gm.cm²)
- (13) P_{rot} - the rotation period (days) of the star (here taken to be independent of depth, so that it is an ‘eigenvalue’, like C above)
- (14) ϕ - the centrifugal-gravitational potential (ergs).
- (15) ϕ_s - the potential at the surface, minus the potential on the L1 surface (ergs); also an ‘eigenvalue’.
- (16) X_{14} - fractional abundance by mass of ^{14}N .
- (17) H_{orb} - the orbital angular momentum (10^{50} gm cm²/sec); also an ‘eigenvalue’.
- (18) e - the eccentricity: also an ‘eigenvalue’.
- (19) ξ - the flux of mass towards or away from the other star (formerly F) (10^{33} gm/sec); a function of depth, but zero below the L1 surface.
- (20) M_{B} - the total mass of the binary; depleted by wind in either or both stars, but not by mass transfer between the stars. Also an ‘eigenvalue’.
- (21) X_{24} - fractional abundance by mass of ^{24}Mg .

Variable (16) is left blank, available for some other function. In the TWIN mode, variables (25) to (40) are the same as (1) to (16), but for the companion star, while variables (17) to (24) are reserved for *binary* parameters.

11 The difference equations

(See Writeup, section 1.5 (p.9) for more explanation).

$$(1) \sigma_{k+1/2}(X_{k+1} - X_k) - \sigma_{k-1/2}(X_k - X_{k-1}) = (\dot{X}_k + R_{\text{nuc},k})m'_k - (X_{k+1} - X_k)[\dot{m}_k] + (X_k - X_{k-1})[-\dot{m}_{k+1}]$$

(2) – (5) similar to (1), for different composition variables: $X4, X12, X16, X20$.

$$(6) \log P_{k+1} - \log P_k = -(Am')_{k+1/2}$$

$$(7) r_{k+1}^2 - r_k^2 = (m'/2\pi\rho r)_{k+1/2}$$

$$(8) \log T_{k+1} - \log T_k = -(\nabla Am')_{k+1/2}$$

$$(9) L_{k+1} - L_k = (m'E_1)_{k+1/2} + (m'E_2)_k[\dot{m}_k] - (m'E_2)_{k+1}[-\dot{m}_{k+1}]$$

$$(10) m_{k+1}^{2/3} - m_k^{2/3} = (2m'/3m^{1/3})_{k+1/2}$$

The additional equations associated with the ML/MB/TF additions are

$$(11) I_{k+1} - I_k = (2m'r^2/3)_{k+1/2}$$

$$(12) \phi_{k+1} - \phi_k = (Gmm'/4\pi r^4\rho)_{k+1/2}$$

$$(19) \xi_{k+1} - \xi_k = \text{CMT} \times (\sqrt{[2\phi_s]}/r m')_{k+1/2}, \quad \text{if } \phi > 0; = 0 \text{ otherwise.}$$

12 The boundary conditions

(1a, 2a, 3a, 4a, 5a, 1b, 2b, 3b, 4b, 5b)

$$\sigma_{k\pm 1/2}(X_k - X_{k\pm 1}) = (\dot{X}_k + R_{\text{nuc},k}) \cdot (m_k - m_{k\pm 1})$$

making 10 such equations in all.

At the surface ($\mathbf{K} = 1$):

$$(6a) \quad dM/dt = -\text{CML} \cdot |\dot{M}_{\text{DDW}}(r, m, L, P_{\text{rot}})| - \text{CMJ} \cdot |\dot{M}_{\text{JNH}}| - \text{CMR} \cdot 1.3 \times 10^{-5} Lm / |E_B| - \text{CMS} \cdot [\ln(R/R_L)]^3 - \text{CMT} \cdot \xi + \text{CMI} \cdot M$$

$$(7c) \quad L = \pi a c r^2 T^4$$

$$(8c) \quad P\kappa/g = 2/3 \quad - \quad \text{more or less}$$

$$(9c) \quad \phi = \text{surface value of potential}$$

$$(10c) \quad d(I\Omega)/dt = \dots, \text{ the rate of change of angular momentum of the star, carried away by stellar wind } |\dot{M}_{\text{DDW}}| \text{ or lost to the orbit by tidal friction } (\Omega \equiv 2\pi/P_{\text{rot}})$$

$$(11c) \quad \phi_s = \phi$$

$$(17c) \quad dH_{\text{orb}}/dt = \dots, \text{ rate of change of } \textit{orbital} \text{ angular momentum, including tidal friction which exchanges ang. mom. between spin and orbit}$$

$$(18c) \quad de/dt = \dots, \text{ rate of circularisation due to tidal friction}$$

$$(20c) \quad dM_B/dt = \text{sum of the winds from both stars; } M_B \text{ is the } \textit{binary} \text{ mass}$$

At the centre ($\mathbf{K} = \mathbf{KH}$), (actually one meshpoint from the centre):

$$(6d) \quad m = 0$$

$$(7d) \quad L = 0$$

$$(8d) \quad r = 0$$

$$(9d) \quad I = 0$$

$$(19d) \quad \xi = 0$$