The Unification I Think Of

Mon. Not. R. astr. Soc. (1968) **138**, 495–525.

THE GRAVO-THERMAL CATASTROPHE IN ISOTHERMAL SPHERES AND THE ONSET OF RED-GIANT STRUCTURE FOR STELLAR SYSTEMS

D. Lynden-Bell and Roger Wood

This study of the thermodynamics of self-gravitating spheres gives insight on the evolution and the final fate of stellar systems. It also helps in the understanding of some well known phenomena in stellar evolution. It is emphasized that these results prove that the escape of stars from a cluster is not necessary for its evolution but rather that extended systems naturally grow a core–halo structure reminiscent of the internal constitution of a red-giant star.

Stellar Evolution

Stephen Justham

UNIVERSITY OF OXFORD

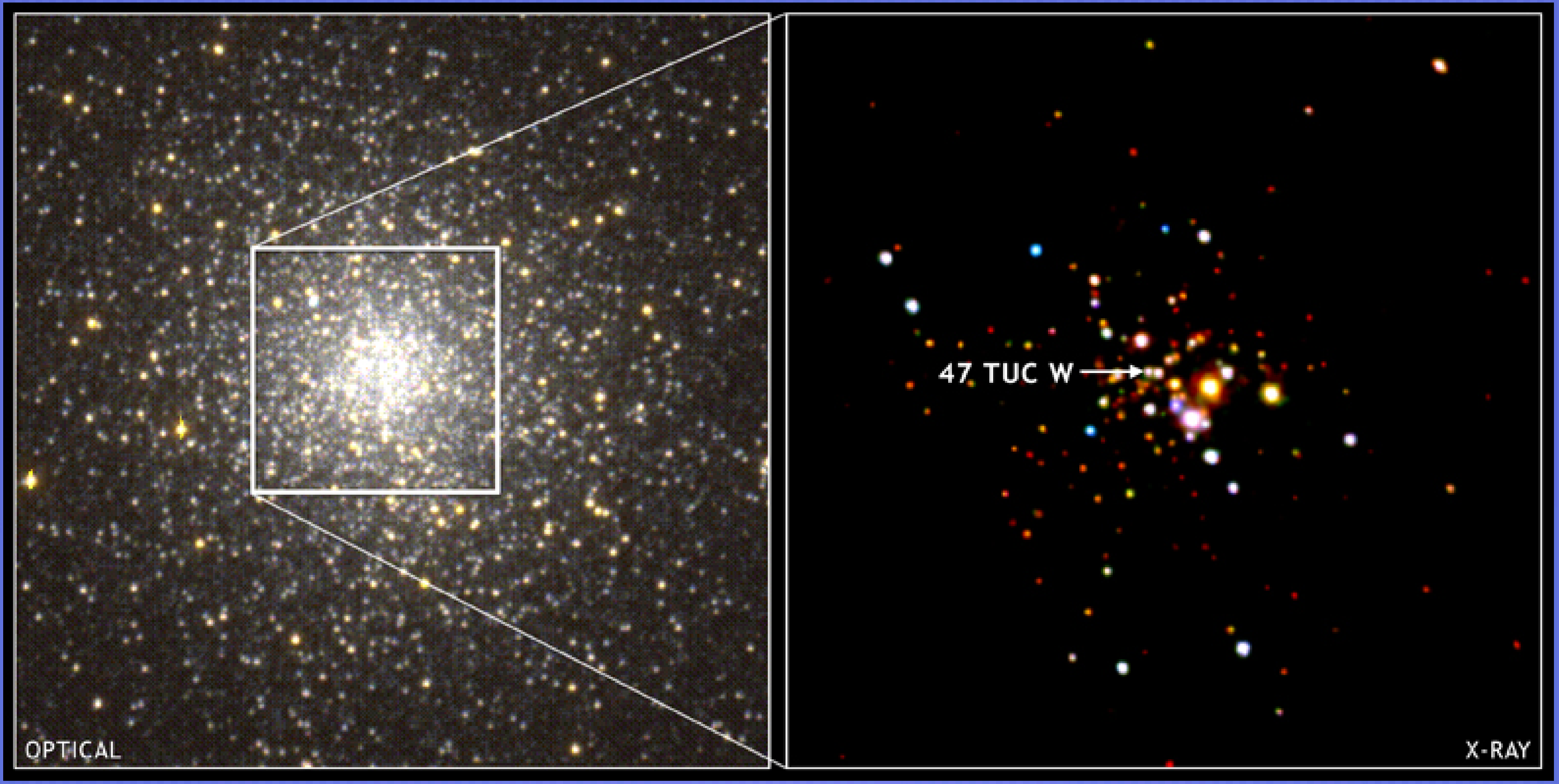
Contents & Intent

- Remind non-stellar people of relevant background & terminology
- Indicate the current state-of-the-art
- Suggest where we should be travelling towards
    - (Robustness, robustness, robustness, new physics, robustness, hydro, robustness....)
- Briefly: wonder whether we could meet more and travel less

Why do stars/binaries matter?

- Even if you are a dynamicist, most of the information about the cluster (i.e. most of the testability) is from the stellar light.
- In a GC, a large fraction of stars will have interacted.
- If you want to use X-ray/UV/radio data, then the compact binaries matter.
- The most interesting individual objects tend to have had the most complicated evolutionary paths.
- Even if you don't care about any of the above, then mass loss (gradual or impulsive) and binary orbits matter to the dynamics.

Why do stars/binaries matter?



47 Tuc in Optical & X-rays (Chandra; Grindlay & Heinke)

Evolving Stars

What I mean by “Stellar Evolution Code”

- Often “Henyey” or “Henyey-type” codes.
- Solve a set of highly non-linear coupled partial differential equations with sometimes pathologically difficult physical coefficients (e.g. from the equation of state).
- Apply boundary conditions at the centre and surface.
- Assume linearity, do some linear algebra to find the corrections from the previous timestep that produce the current solution.
- Unsurprisingly, convergence sometimes fails.
- Mostly legacy codes; hardly anyone really understands one well.

What I mean by "Binary Evolution Code"

- Minimum – a stellar evolution code as before, with:
  - A way to allow binary mass loss/gain (modified BC).
  - Circular orbits; a point mass companion; Roche-potentials.
  - Angular momentum dealt with (winds, RLOF, Braking, GWR).

- TWIN is better for Algol-type systems:
  - Evolves both stars simultaneously.
  - Mass can be transferred directly between components.
  - Does NOT deal with contact binaries (no-one can).

Binary interactions:

- What spherically symmetric Henyey-type evolution codes can do:
  - Roche-lobe overflow.
  - Potentially some tidal effects, effects of rapid rotation.
- What they can't do:
  - Common-envelope evolution (vastly, vastly important).
  - Other sudden changes in binary parameters (e.g. SNae).
  - Mergers & other 3d hydrodynamic processes.

- So even with a "full" binary code, you need special-case subroutines.

Calculation Speed

Mon. Not. R. astr. Soc. (1971) **151**, 351–364.

THE EVOLUTION OF LOW MASS STARS

Peter P. Eggleton

We find that typically 60 models are needed between the main sequence and the base of the giant branch, and a further 60 from there to the top of the giant branch, when helium burning sets in. This is dictated by the fact that roughly equal amounts of fuel are consumed in these two stages. Each model demands one or two iterations, depending on the accuracy specified, and for a 100-point mesh each iteration requires ~ 25 s on an IBM 360/44 machine. Thus an evolutionary sequence requires about 60 min from the main sequence to the helium flash, if fairly low accuracy (~ 1 per cent) is demanded for each model.

Today, we typically use more meshpoints, demand higher accuracy and shorter timesteps. 5 mins for full stellar evolution fastish today.

(? <20s equivalent of the above on my >2-year-old laptop?)

Note: Calculation Speed

- Bare calculation speed is misleading.
  - Evolving a star robustly in an hour will mean your N-body cluster can actually run (slowly!).
  - If you evolve each binary in 30 seconds, but 90% crash (most of the interesting ones) then there’s not much point.

Full Stellar/Binary Codes

- Eggleton & derivatives (STAR / ds2000 / TWIN / EZ)
  - Many people involved in MODEST; EFT & HPT fits.
- Kippenhahn code & descendants (Langer, Podsiadlowski, ...?)
- Paczynski code & descendants (Brussels, must be others....?)
- Kovetz & Prialnik & Yaron (a rewritten/new code?)
- Mazzitelli & variants (Kolb, Schenker for CVs; convection studies?)
- Tycho (Arnett), Geneva, Yale-Yonsei, Baraffe (low-mass)
- Padova (Girardi, Chiosi....), Meynet & Maeder (rotating single)
- Icko Iben <u>surely</u> must have written a code... or used another?
- MESA (forthcoming – Bill Paxton & friends).

Analytics: fast and robust.

- We know the answer (well, within the limits of our ability) for single-star evolutions. Why bother to recalculate them?
- Fit previously-know results with analytic formulae.
- Extremely fast (essentially instant results).
- Even better: very numerically robust.
- Excellent for single stars.
- EFT & HPT are the best. (HPT 00 = SSE)

HPT example:

The luminosity at the end of the MS is approximated by

$$L_{\mathrm{TMS}} = \frac{a_{11}M^3 + a_{12}M^4 + a_{13}M^{a_{16}+1.8}}{a_{14} + a_{15}M^5 + M^{a_{16}}}, \tag{8}$$

with $a_{16} \approx 7.2$. This proved fairly straightforward to fit, but the behaviour of R_{TMS} is not so smooth and thus requires a more complicated function in order to fit it continuously. The resulting fit is

$$R_{\mathrm{TMS}} = \frac{a_{18} + a_{19}M^{a_{21}}}{a_{20} + M^{a_{22}}} \quad M \leq a_{17} \tag{9a}$$

$$R_{\mathrm{TMS}} = \frac{c_1 M^3 + a_{23}M^{a_{26}} + a_{24}M^{a_{26}+1.5}}{a_{25} + M^5} M \geq M_*, \tag{9b}$$

with straight-line interpolation to connect equations (9a) and (9b) between the end-points, where

$M_* = a_{17} + 0.1, \quad 1.4 \leq a_{17} \leq 1.6,$

HPT example:

The luminosity at the base of the AGB (or the end of CHeB) is given by

$$L_{\mathrm{BAGB}} = \begin{cases} \dfrac{b_{29} M^{b_{30}}}{1 + \alpha_3 \exp 15(M - M_{\mathrm{HeF}})} & M < M_{\mathrm{HeF}} \\[2ex] \dfrac{b_{31} + b_{32} M^{b_{33}+1.8}}{b_{34} + M^{b_{33}}} & M \geq M_{\mathrm{HeF}} \end{cases} \tag{56}$$

with $\alpha_3 = [b_{29} M_{\mathrm{HeF}}^{b_{30}} - L_{\mathrm{BAGB}}(M_{\mathrm{HeF}})]/L_{\mathrm{BAGB}}(M_{\mathrm{HeF}})$. The radius at the BAGB is simply $R_{\mathrm{AGB}}(M, L_{\mathrm{BAGB}})$, as given by equation (74).

The lifetime of CHeB is given by

$$t_{\mathrm{He}} = \begin{cases} \{b_{39} + [t_{\mathrm{HeMS}}(M_{\mathrm{c}}) - b_{39}](1 - \mu)^{b_{40}}\} & \\ \quad \times [1 + \alpha_4 \exp 15(M - M_{\mathrm{HeF}})] & M < M_{\mathrm{HeF}} \\[2ex] t_{\mathrm{BGB}} \dfrac{b_{41} M^{b_{42}} + b_{43} M^5}{b_{44} + M^5} & M \geq M_{\mathrm{HeF}} \end{cases} \tag{57}$$

HPT example:

The Z-dependence of the coefficients a_n and b_n is given here.
Unless otherwise stated,

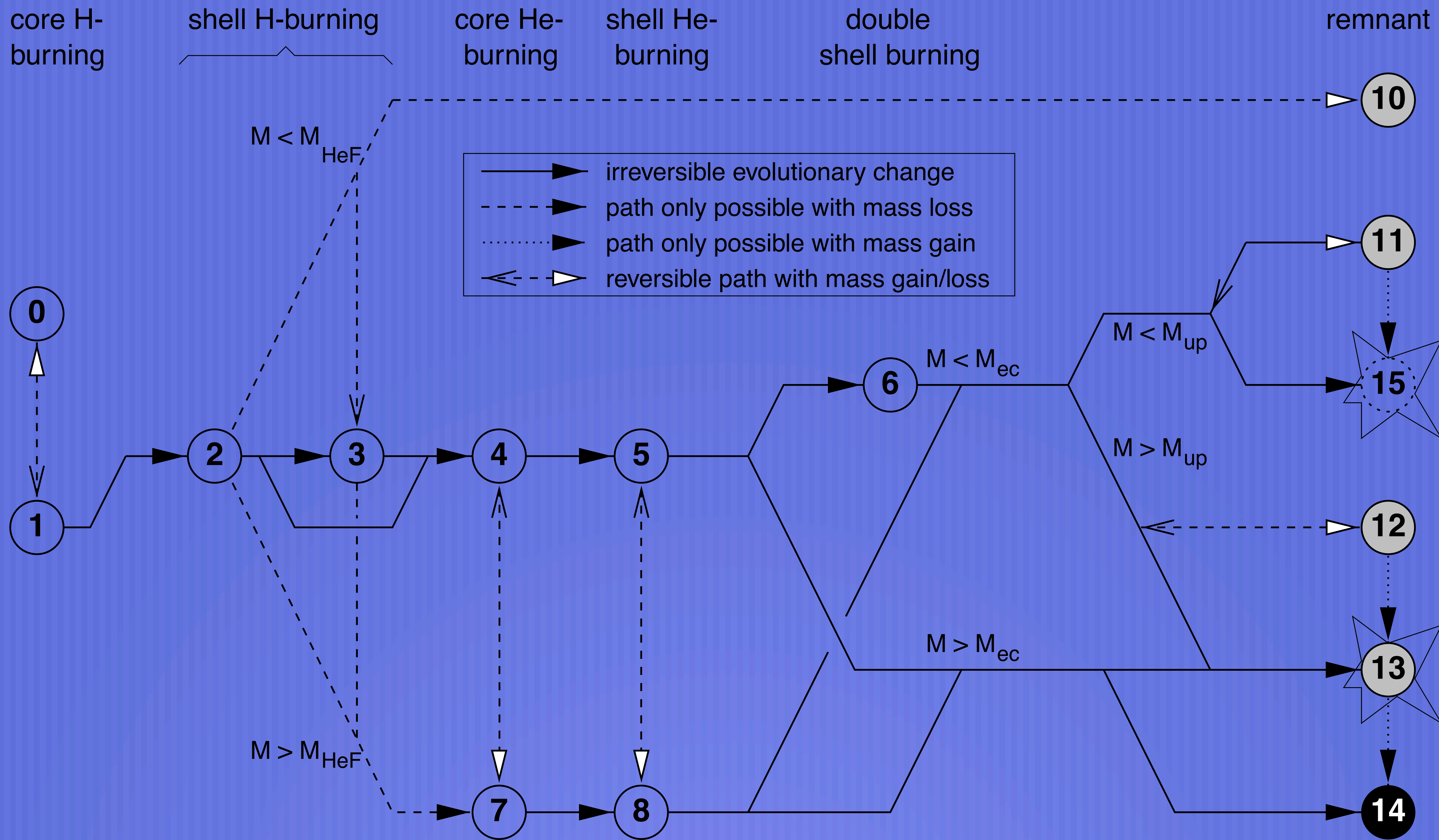
$a_n = \alpha + \beta\zeta + \gamma\zeta^2 + \eta\zeta^3 + \mu\zeta^4$,

and similarly for b_n, where

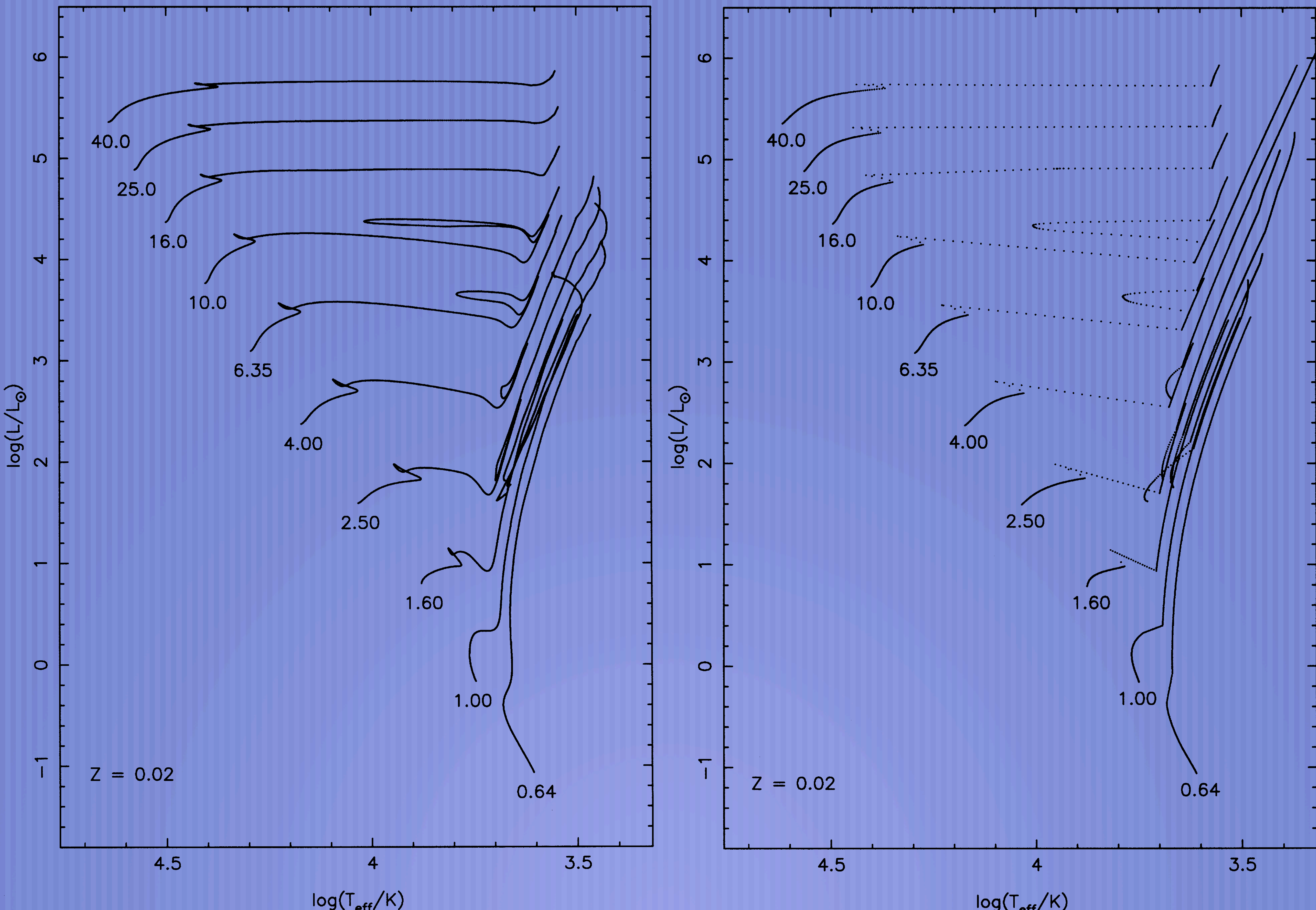
$\zeta = \log(Z/0.02)$.

	α	β	γ	η	μ
a'_{18}	2.187715(−1)	−2.154437(+0)	−3.768678(+0)	−1.975518(+0)	−3.021475(−1)
a'_{19}	1.466440(+0)	1.839725(+0)	6.442199(+0)	4.023635(+0)	6.957529(−1)
a_{20}	2.652091(+1)	8.178458(+1)	1.156058(+2)	7.633811(+1)	1.950698(+1)
a_{21}	1.472103(+0)	−2.947609(+0)	−3.312828(+0)	−9.945065(−1)	
a_{22}	3.071048(+0)	−5.679941(+0)	−9.745523(+0)	−3.594543(+0)	
a_{23}	2.617890(+0)	1.019135(+0)	−3.292551(−2)	−7.445123(−2)	
a_{24}	1.075567(−2)	1.773287(−2)	9.610479(−3)	1.732469(−3)	
a_{25}	1.476246(+0)	1.899331(+0)	1.195010(+0)	3.035051(−1)	
a_{26}	5.502535(+0)	−6.601663(−2)	9.968707(−2)	3.599801(−2)	

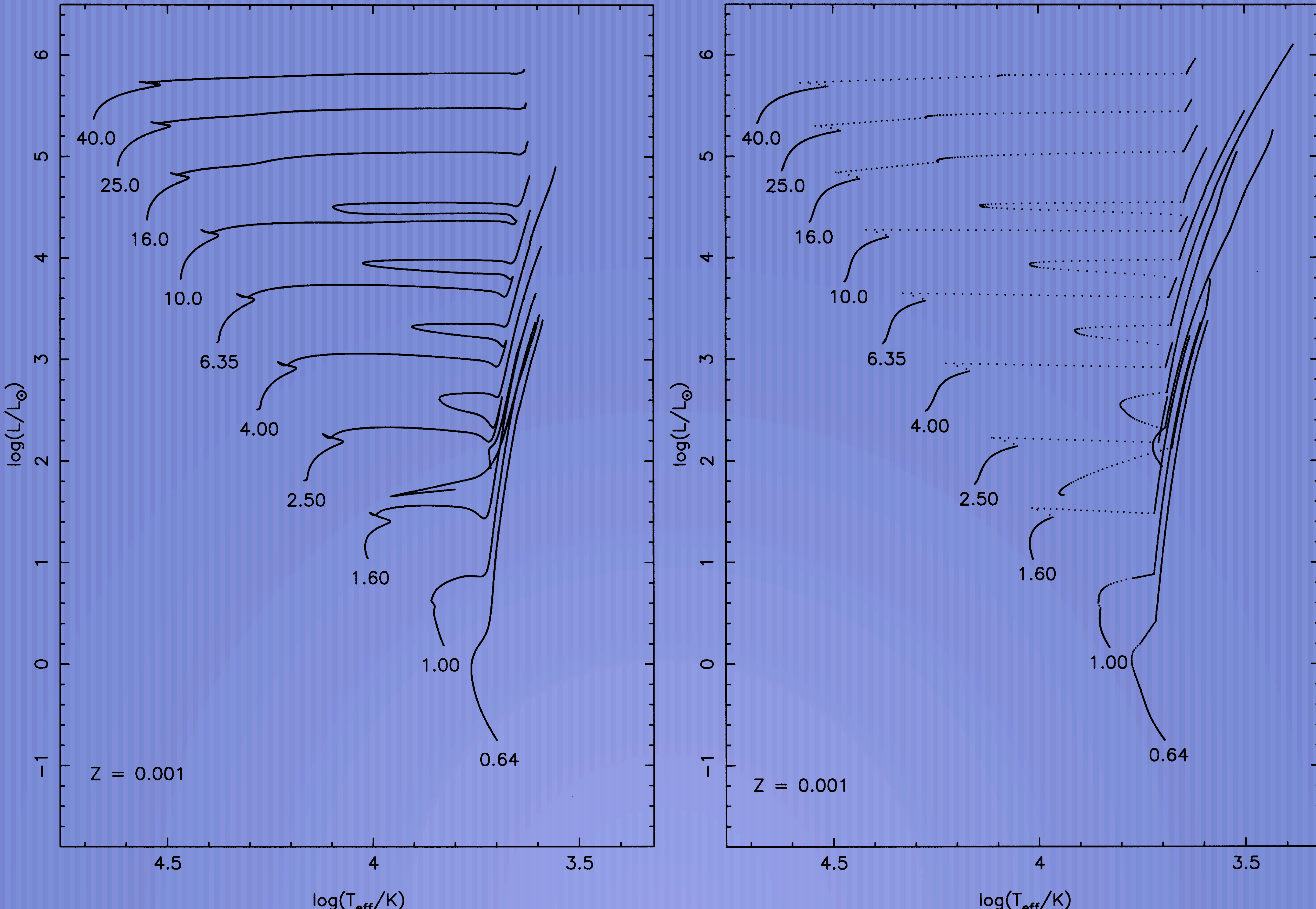
$\log a_{17} = \max[0.097 - 0.1072(\sigma + 3),\ \max\{0.097,\ \min[0.1461, 0.1461 + 0.1237(\sigma + 2)]\}]$
$a_{18} = a'_{18} a_{20}$
$a_{19} = a'_{19} a_{20}$



0 = main sequence $M < 0.7\ M_\odot$
1 = main sequence $M > 0.7\ M_\odot$
2 = Herstzsprung gap / subgiant
3 = first-ascent red giant
4 = horizontal branch / helium-burning giant
5 = early asymptotic giant / red supergiant
6 = thermally pulsating asymptotic giant
7 = naked helium main sequence
8 = naked helium (sub) giant
10 = helium white dwarf
11 = carbon/oxygen white dwarf
12 = oxygen/neon white dwarf
13 = neutron star
14 = black hole
15 = no stellar remnant



Hurley, Pols & Tout single-star fitting formulae (2000).

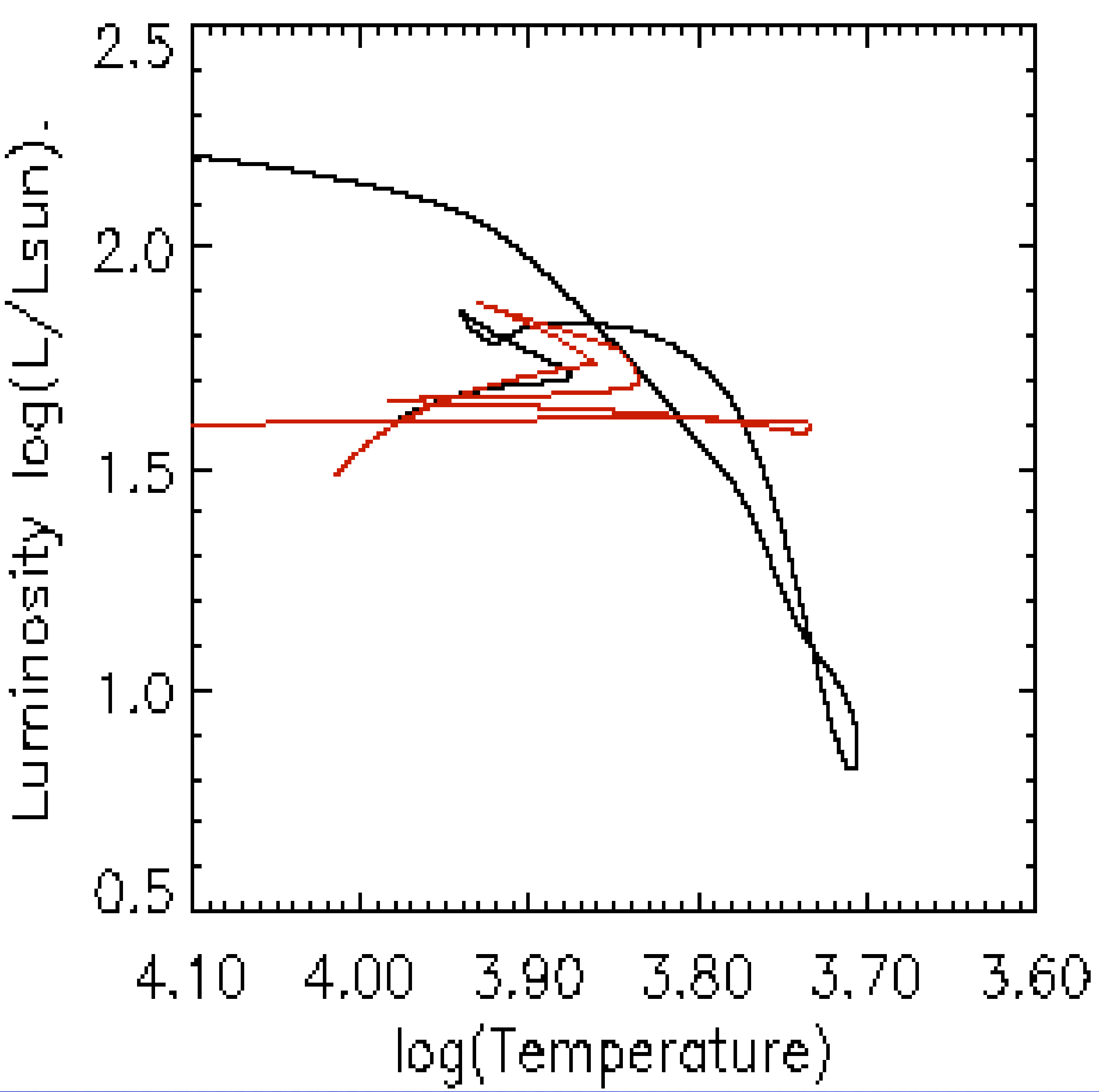


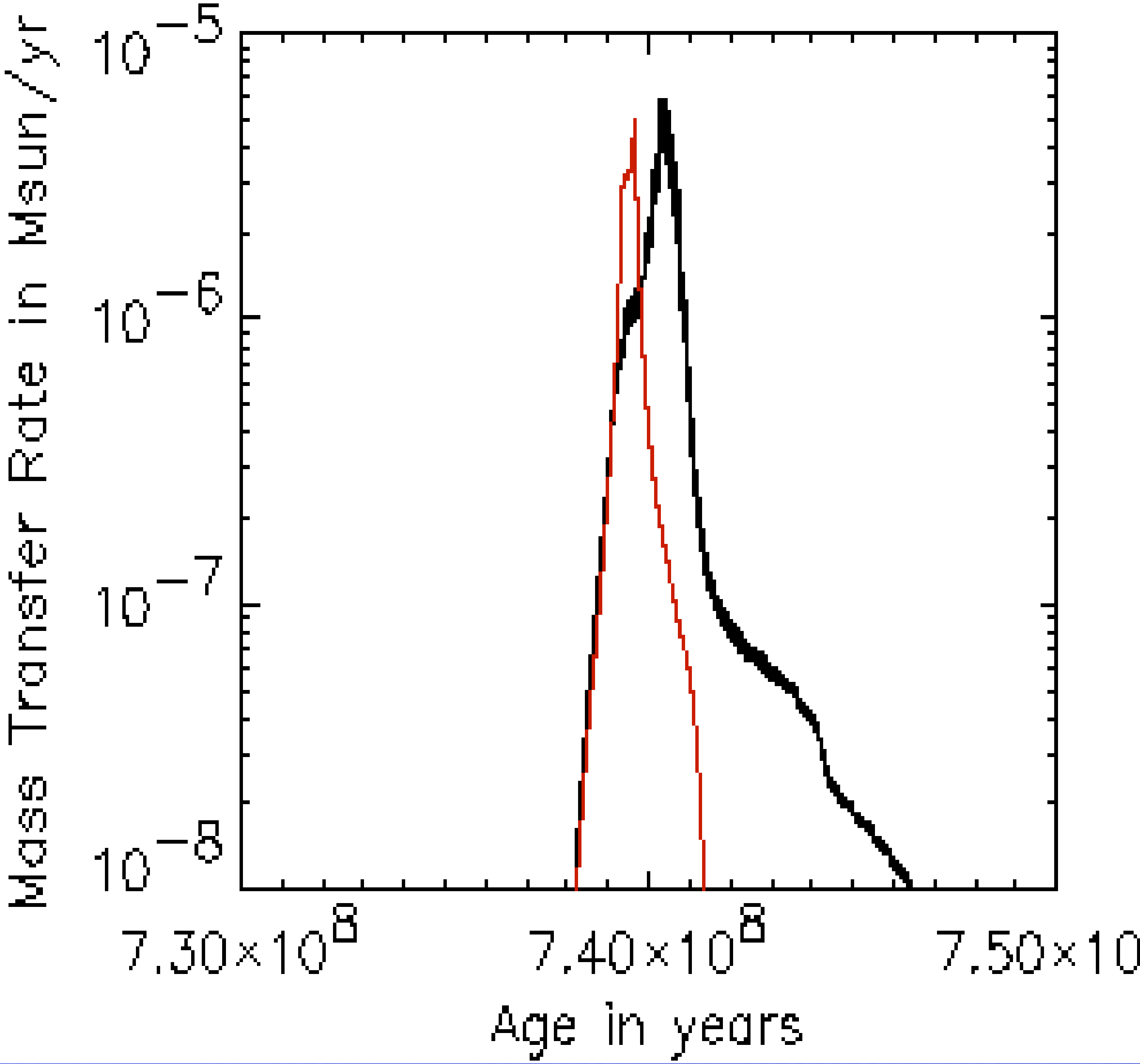
More Hurley, Pols & Tout single-star fitting formulae (2000).

Fitting formulae: disadvantages

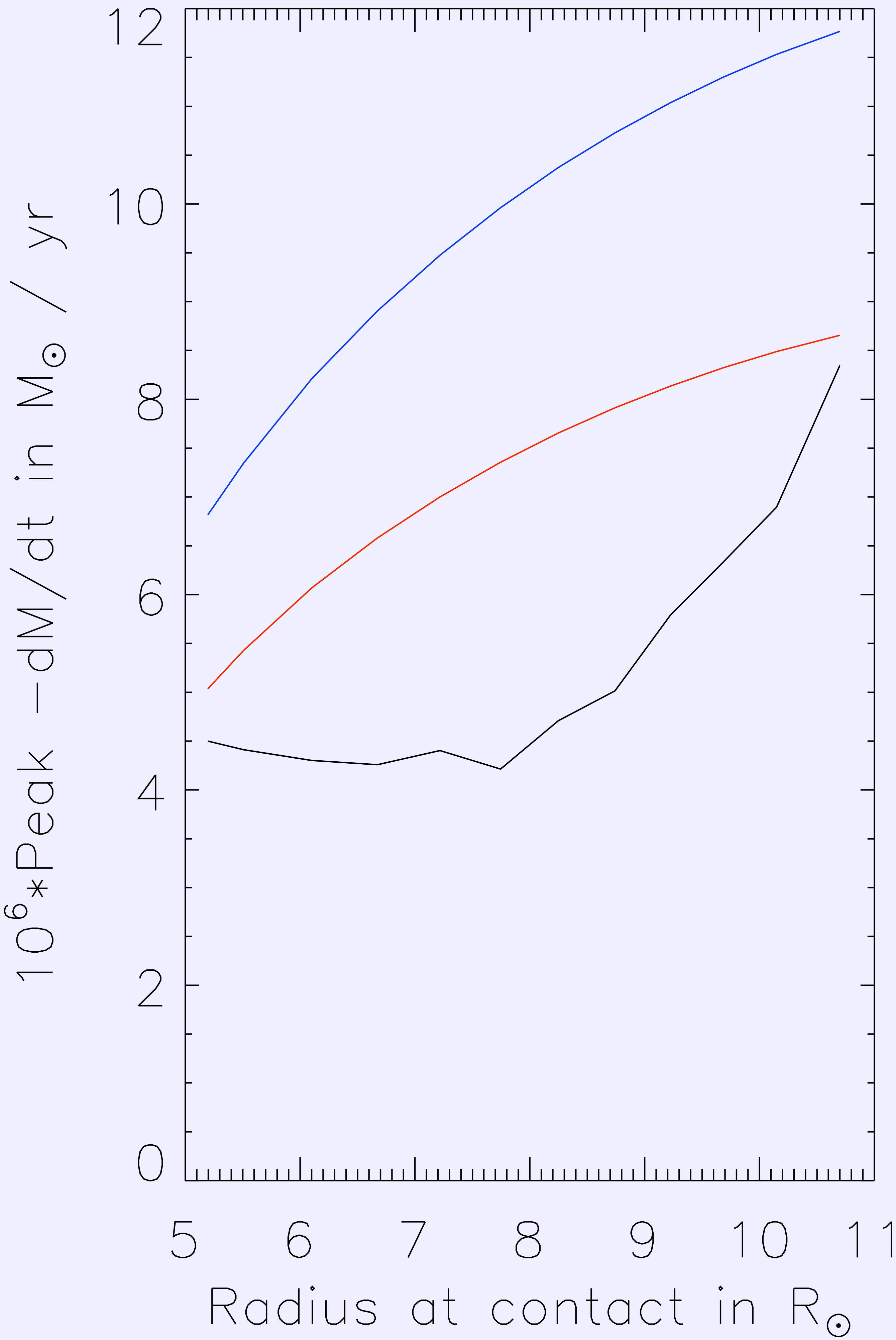
- Effort to make; you can't easily try changing your stellar physics, or adding new effects. May have to be completely re-made as stellar physics improves.
- The best we have (HPT 2000) are not guaranteed to better than 5%.
- The single star fitting-formulae need to be supplemented with recipes to deal with binary mass transfer....
  - HPT 00 (SSE) becomes HTP 02 = BSE (SSE + binary recipes)
  - Results sometimes good, sometimes highly approximate.
  - Arguably, the recipes become worse in more interesting cases.

Tracks: disadvantages





2.35 $M_\odot$ donor, 0.95 $M_\odot$ accretor, 17.9h initial orbital period.



3 $M_\odot$ donor, 1.4 $M_\odot$ accretor.

Note: this tells you that the mass transfer phase will be wrong, but actually the remnant evolution in this case is okay. In other situations, this would be reversed....

Tracks: disadvantages

- Effort to make; may have to be re-made if stellar physics improves.
- The best we have (HPT 2000) are not guaranteed to better than 5%.
- Binary recipes are sometimes excellent, sometimes highly approximate.
- Two extremes:
  - (1) The true answer is uncertain. Live with it.
  - (2) Disdain fitting formulae for not being perfect.
- The middle way: use analytics (or stored exact results) when you can. Perform full calculations when you need to.

Formula-based Binary codes:

- SSE & BSE (Hurley, Pols, & Tout)
  - StarTrack (Kalogera, Belczynski, ...), BiSEPS (Willems, Kolb,...)
  - Jarrod Hurley? Probably others too....
- Other:
  - SeBa (SPZ, ...), Moscow (Yungleson, Tutukov, ...), ?
- Alternatives:
  - Interpolation between tracks saved to disk (e.g. Brussels)
  - Combine SSE with detailed tracks for specific classes of system (e.g. Podsiadlowski, Rappaport, Pfahl & Han....)

Things we know we can't do.

- Stellar winds (<u>very</u> important for massive stars, possibly GC HBs)
- Convection (overshooting; enhanced mixing; better than fixed MLT?)
- Contact binaries (usually ignored...)
- Common-envelope evolution
- Mass transfer in eccentric systems (but see Willems, Sepinski et al.)
- Mergers (though see Lombardi, Glebbeek, Sills)
- Angular momentum (magnetic braking; transfer in RLOF; rapid rotators)
- Tidal spin-up / synchronisation / eccentricity damping / heating.
- Accretion (efficiency and mass loss, evolution of accreting stars, ...)
- Donor irradiation
- "Supermassive" stars (IMBHs? Pulsations / eruptions / wind-loss?)
- Supernovae (core-collapse kicks?; e-capture; what makes an SN Ia?; AIC?)
- Initial conditions! (a, P, M, q).

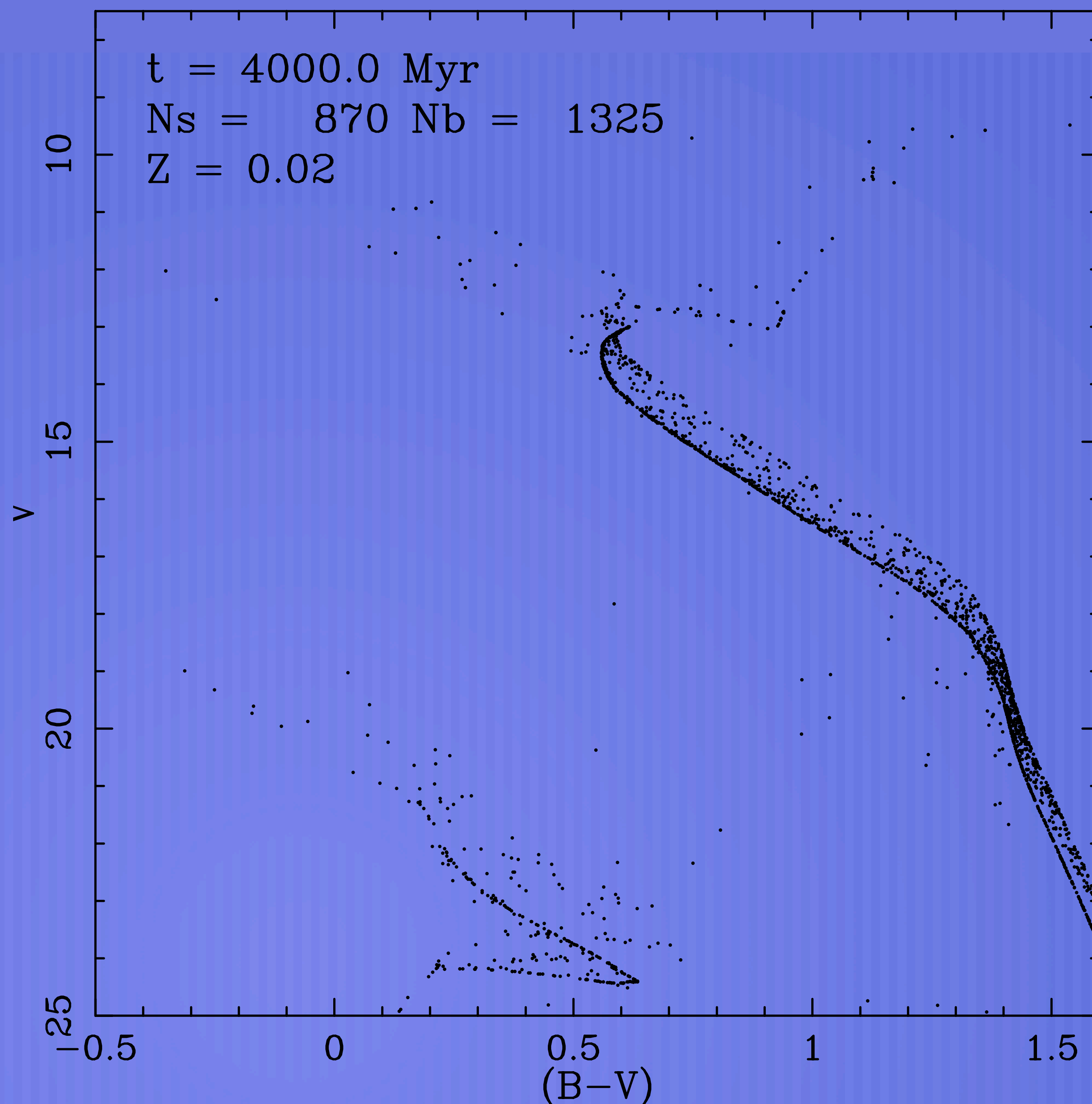
What has been done?

A complete *N*-body model of the old open cluster M67

Jarrod R. Hurley,[1,2★] Onno R. Pols,[3] Sverre J. Aarseth[4] and Christopher A. Tout[4]

Here, we present the first truly direct *N*-body model for M67, evolved from zero age to 4 Gyr taking full account of cluster dynamics as well as stellar and binary evolution. Our preferred model starts with 36 000 stars (12 000 single stars and 12 000 binaries) and a total mass of nearly 19 000 $M_\odot$, placed in a Galactic tidal field at 8.0 kpc from the Galactic Centre.

BSE & NBody4



Formation and evolution of compact binaries in globular clusters – I. Binaries with white dwarfs

N. Ivanova,[1★] C. O. Heinke,[2†] F. A. Rasio,[2] R. E. Taam,[2] K. Belczynski[3‡]
and J. Fregeau[2]

**Formation and evolution of compact binaries in globular clusters:
II. Binaries with neutron stars.**

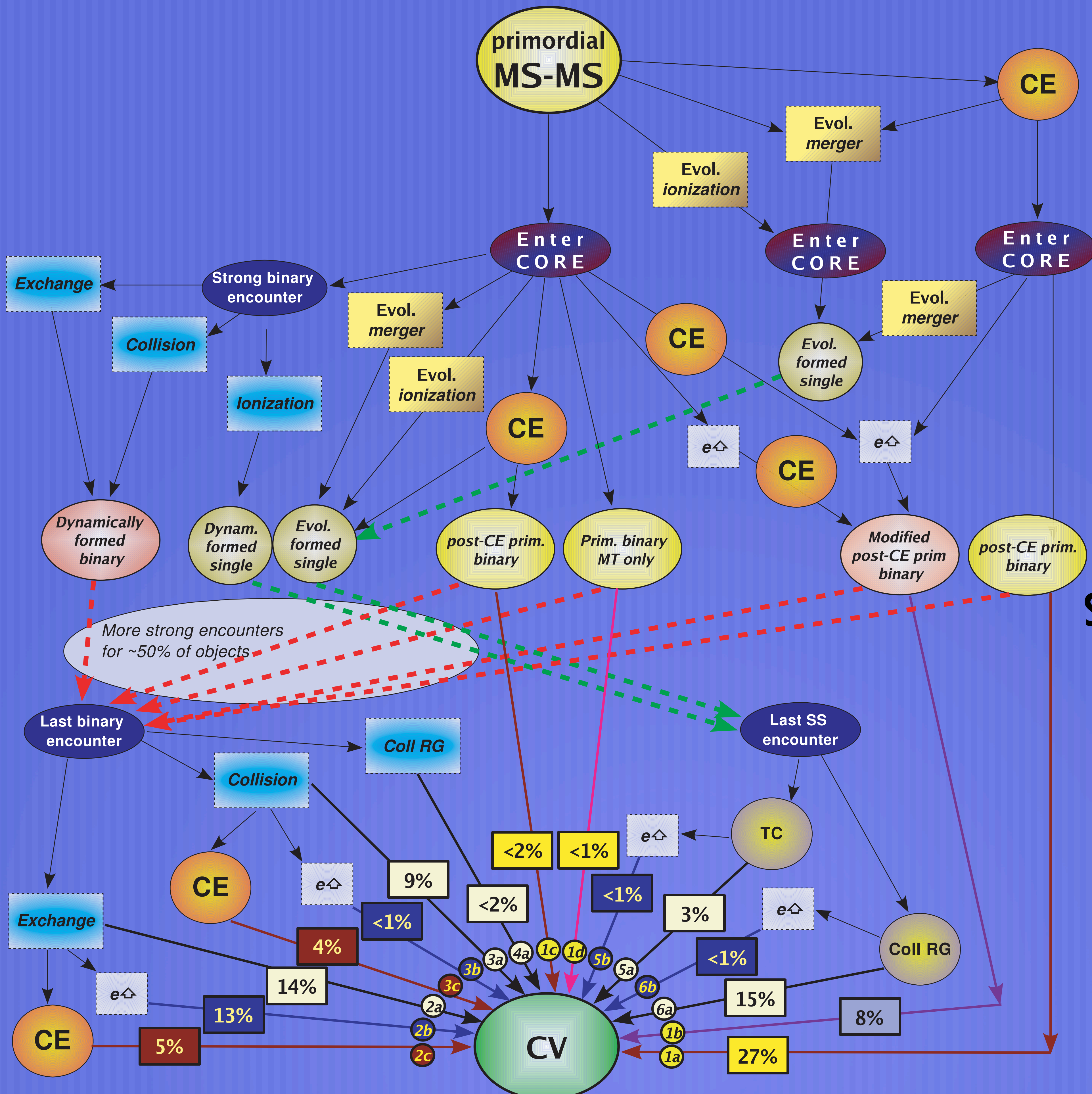
N. Ivanova [1⋆], C. Heinke[2†], F. A. Rasio[2], K. Belczynski[3‡], & J. Fregeau[2]

StarTrack with some new recipes for dynamical encounters

&

Fewbody integrator, very simple “Monte Carlo” (but not Monte Carlo dynamical code!) treatment of interations.

Uses 10^6 stars.



NB: no supernova kicks!

100 single stars & 100 binaries.

Ross Peter Church
Jesus College

Institute of Astronomy
University of Cambridge

Chapter 5

N-body Models with Binary Stars

This chapter explains the extended interface used to implement binary stellar evolution, calculated with the modified version of `STARS`, in `NBODY6`. A model of a small stellar cluster made with the new code is presented and discussed. This is necessarily only the first step towards the inclusion of full binary evolution in N-body models of clusters but it demonstrates that this goal is achievable by its rudimentary success.

Where are we heading?

Aiming at MODEST goals:

- If we can calculate one full binary evolution in one minute, and can use 20 processors for 24 hours:
  - $60\times24\times20=1440\times20=28800$ binaries.
- Use for binary population synthesis without having to rely on 'analytics' or interpolation formulae.
- This has been done for pseudo-binaries containing only one "evolving" star (mean evolution time $\simeq$4mins, $>10^5$ evolutions).
- In practice, much of that CPU time was repetetive.

The problem is not (really) speed

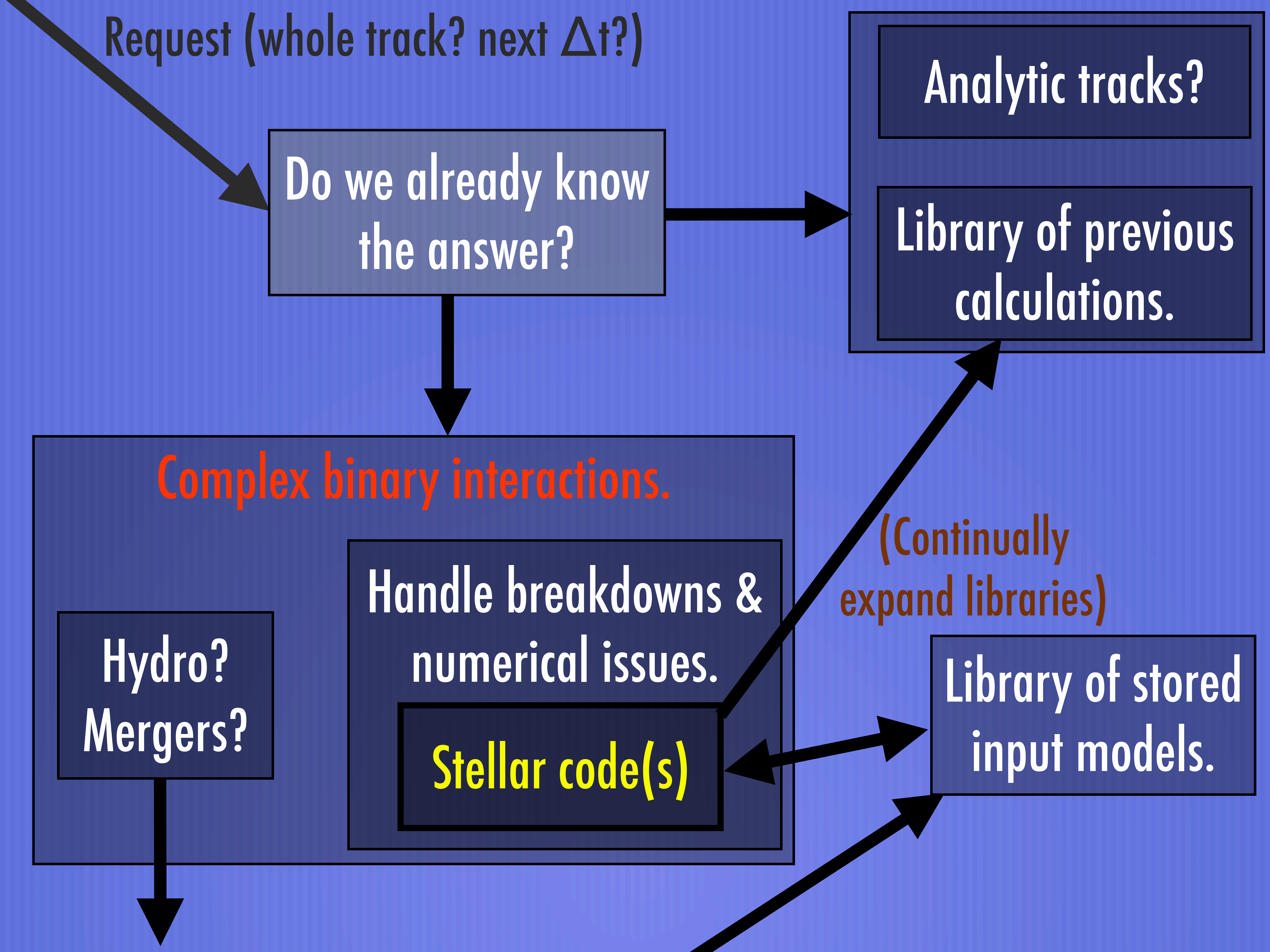
- So if I can calculate $>10^5$ binaries in a day with 20 processors, what's the problem?
- Indeed; it would be even faster if we were efficient about using previous calculations / intermediate models.
- Note, however, that this is only one mass transfer episode, not the full life-history of the binary.
- <u>Reliability.</u> Many of those binaries broke, and needed human interaction. Do you want your N-body code to be permanently sitting and waiting for grad-students/postdocs to sort out the troublesome cases?

Robustness....

- Can we improve on the basic solution method?
  - Techniques from condensed matter theorists? Applied mathematicians?
- Can we at least make the convergence algorithm better?
- We can certainly code-in much of what users do using their brains & experience to get through difficult cases. Eventually...
  - And how many attempts do you want to try in each case? For how long?
  - Also smooth EOS / input physics as much as possible (e.g. MESA).
- We'll probably still need an exception-handling convention.
- Interested in Capri poster by Ofer Yaron!

What should a black box stellar MUSE module look like?

- To specify a previously non-interacting single star:
  - INITIAL: mass, composition
  - CURRENT: age, mass, core mass, core composition, type?
- Keeping 10^6 stars in memory is hard
  - ?1Mb each, more common; 0.1Mb for super-minimalist Eggleton.
- What do we want to know?
  - M, R, dM/dt, L, T, M_c, X_c, Y_c, X, Y, (etc) all as function of time.
  - Sometimes the full structure for input into hydro/merger?
  - Did the star go supernova? What remnant & kick velocity?
- For the general case, we want to be able to input a full structure too. Hence want a flag for “bespoke” vs “off-the-shelf” stars.


Request (whole track? next Δt?)
Do we already know the answer?
Analytic tracks?
Library of previous calculations.
Complex binary interactions.
Handle breakdowns & numerical issues.
Stellar code(s)
Hydro? Mergers?
(Continually expand libraries)
Library of stored input models.

Could we travel less and meet more?

Final Thought

- Split (& Amsterdam, Chicago, ...) are very enjoyable (so far!).
- Air travel is bad for the planet, the cost is bad for our budgets, infrequent meetings are bad for the MUSE project...
- Electronic tools exist for regular meetings.
- The particle physics instrumentalists in my building (and some astro instrumentalists) have regular teleconferences... aren't we theory-instumentalists?

Conclusions: Stability

- Binary evolution is not a fire-and-forget computational problem: they are quick and routine calculations, but they are not easy.
- It would be extremely good if codes were much more robust.
- This is particularly important if you consider that the other problems we wish to address (such as evolving collision products) take rather more effort to converge than most "normal" binaries. Currently we are doing the easy ones.