Binary Populations

Robert Izzard
University of Surrey
(Just south of LHR, west of LGW)

binary_c

Science & Technology Facilities Council

UNIVERSITY OF SURREY

- Binary populations → single and multiple stars
- Properties of a stellar population
- What is and why use population synthesis
  - The (huge) parameter space problem
- Stellar modelling for population synthesis
  - → *binary_c* code
- Lots of examples for you to try

Population Synthesis

What is a “population” ?

“Stellar” populations

From galaxies...

...to stars...

… to planets.

What is a synthesis? One star:

$$\frac{dP}{dm} = -\frac{Gm}{4\pi r^4}$$

$$\frac{dr}{dm} = \frac{1}{4\pi r^2 \rho}$$

$$\frac{dL}{dm} = \epsilon$$

$$\frac{dT}{dm} = -\frac{3}{4ac}\frac{\kappa}{T^3}\frac{F}{(4\pi r^2)^2}$$

Stellar structure equations

Lifetime, mass, radius, luminosity, colours, chemistry, ejecta etc.

Population synthesis: "make" many stars






$$\frac{dP}{dm} = -\frac{Gm}{4\pi r^4}$$

$$\frac{dr}{dm} = \frac{1}{4\pi r^2 \rho}$$

$$\frac{dL}{dm} = \epsilon$$

$$\frac{dT}{dm} = -\frac{3}{4ac}\frac{\kappa}{T^3}\frac{F}{(4\pi r^2)^2}$$

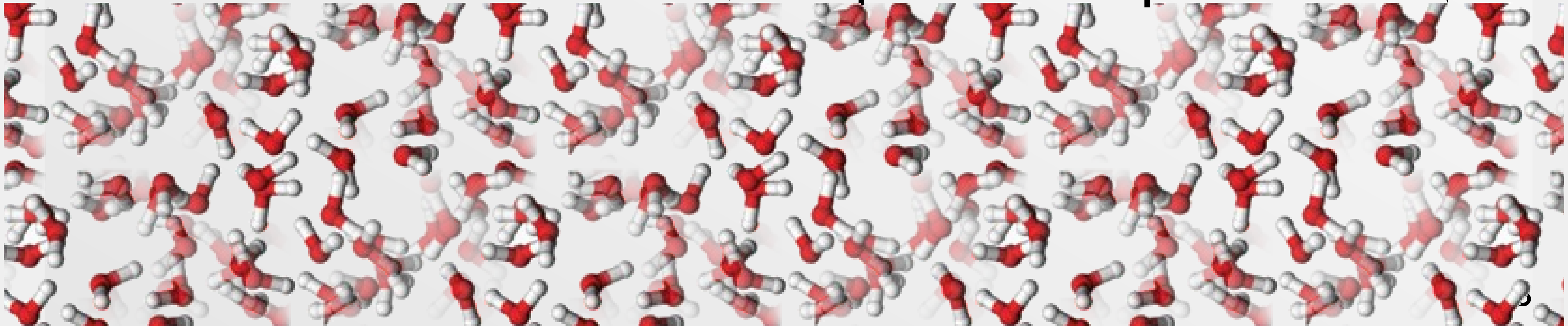
STATISTICS: distributions of lifetimes, masses, radii, luminosities, colours, chemistries, integrated ejecta, orbital properties, etc.

A method to understand stars

- Given **observations** of a stellar population
- Make a **model synthesis**
- Compare the two: involves **statistics** (sorry!) then:
- Make **new predictions** ? Improve the **model** ?
- Improve the **observations** ?

Why are stars special?

- Population synthesis is not unique to astronomy, but it is perhaps more useful than in other sciences. Why?
- Stars are complex objects. While we **cannot experiment** on them we **can model them.**
- Stars are mostly isolated: cf. **atoms** or **molecules**!
- So just add up the properties of stars, both in observations and models, compare and improve.

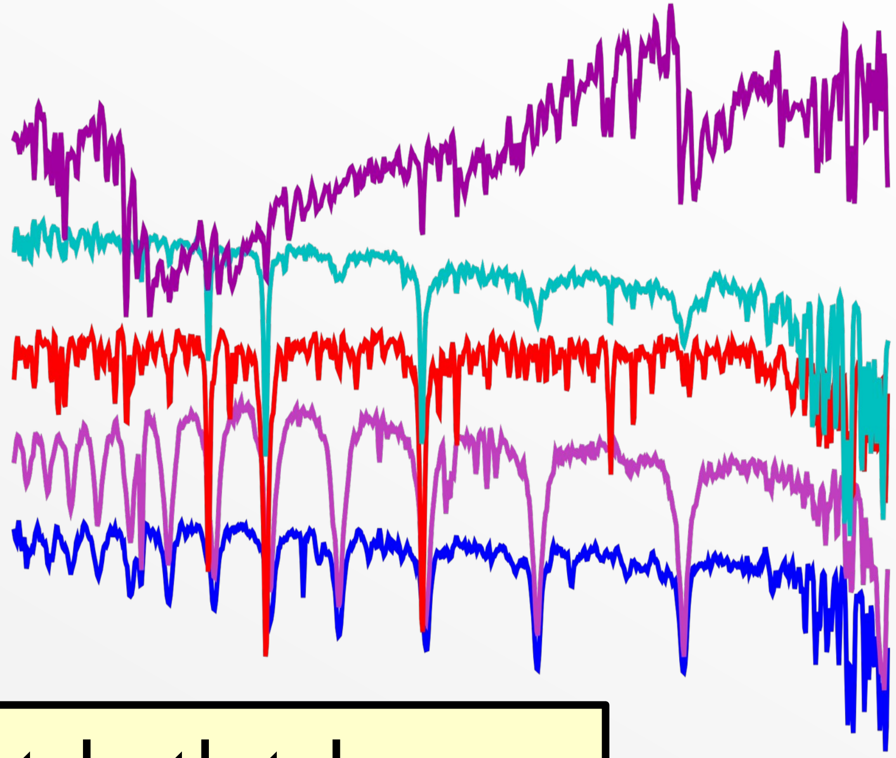


Population Synthesis in Astronomy

- Spectral population synthesis

  vs.

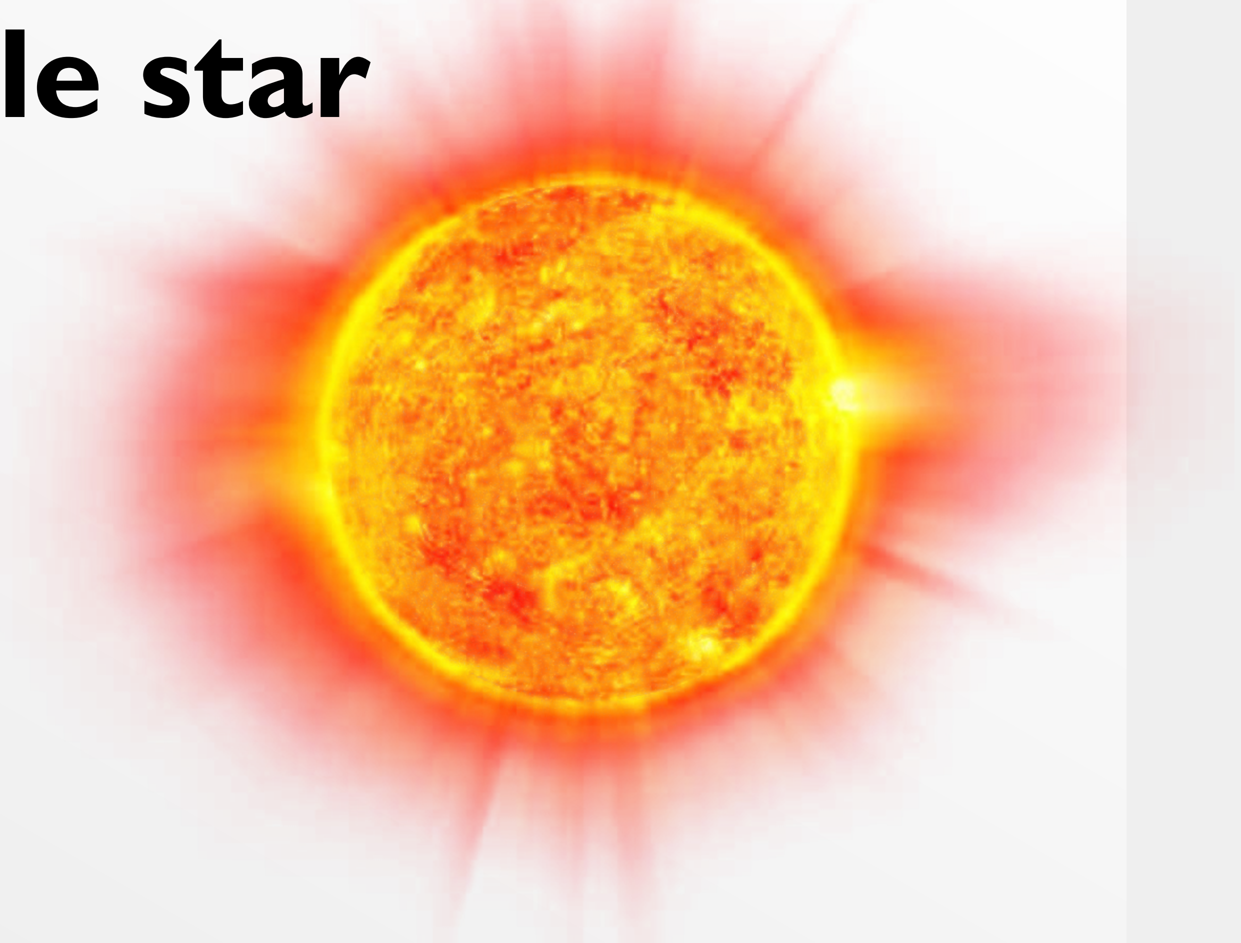
- Evolutionary population synthesis



These are not so different: both take predictions of stellar evolution models, add them up and compare to observations.
We will talk about ***evolutionary population synthesis (of binary stars)***, but the concepts are very similar.

Key parameters of an isolated, single star

- Mass, *M*
- Metallicity, *Z*
- Rotation rate, *v* … ?

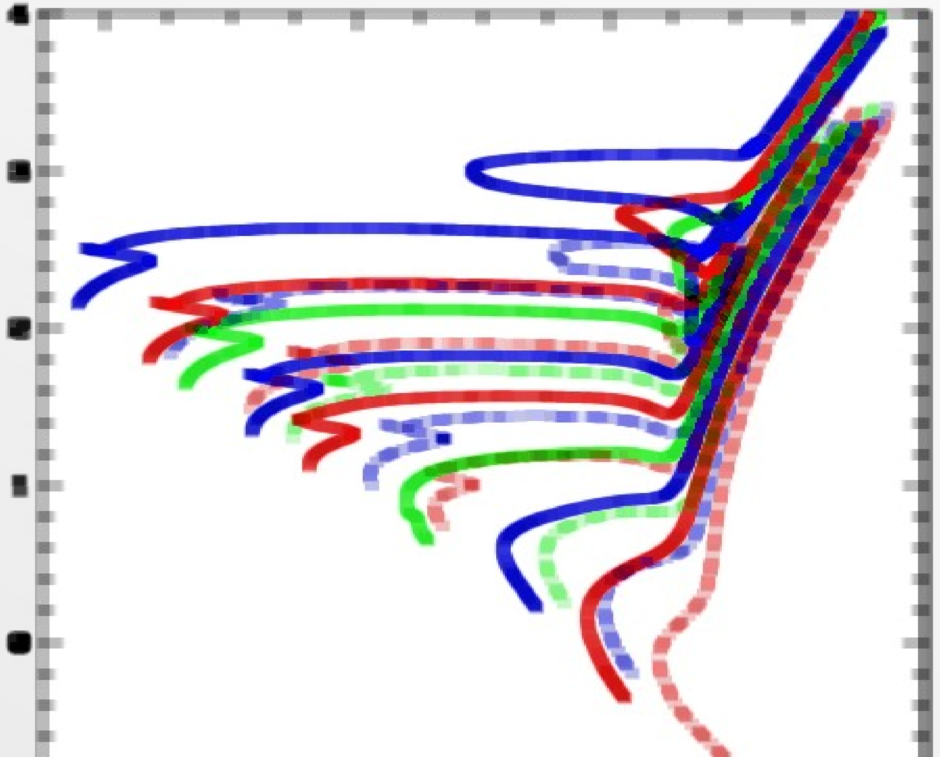


This assumes we have ideal stellar models.
(The other lecturers will help me out here :)

Uncertain parameters in single stellar evolution

- Convection e.g. overshooting, undershooting
- Mass loss rates, esp. AGB and massive stars
- Extra mixing: thermohaline, diffusion, whatever
- Rotational mixing
- Magnetic fields
- Explosion mechanism

  (SN II, Ib/c, Ia, GRB)



Other lectures on these!

A first synthetic population

- Assume fixed metallicity *Z* and neglect rotation
- Fix all other physics input to “best” model
- Starburst: all stars are single and born at t=0
- Single parameter: initial mass *M*

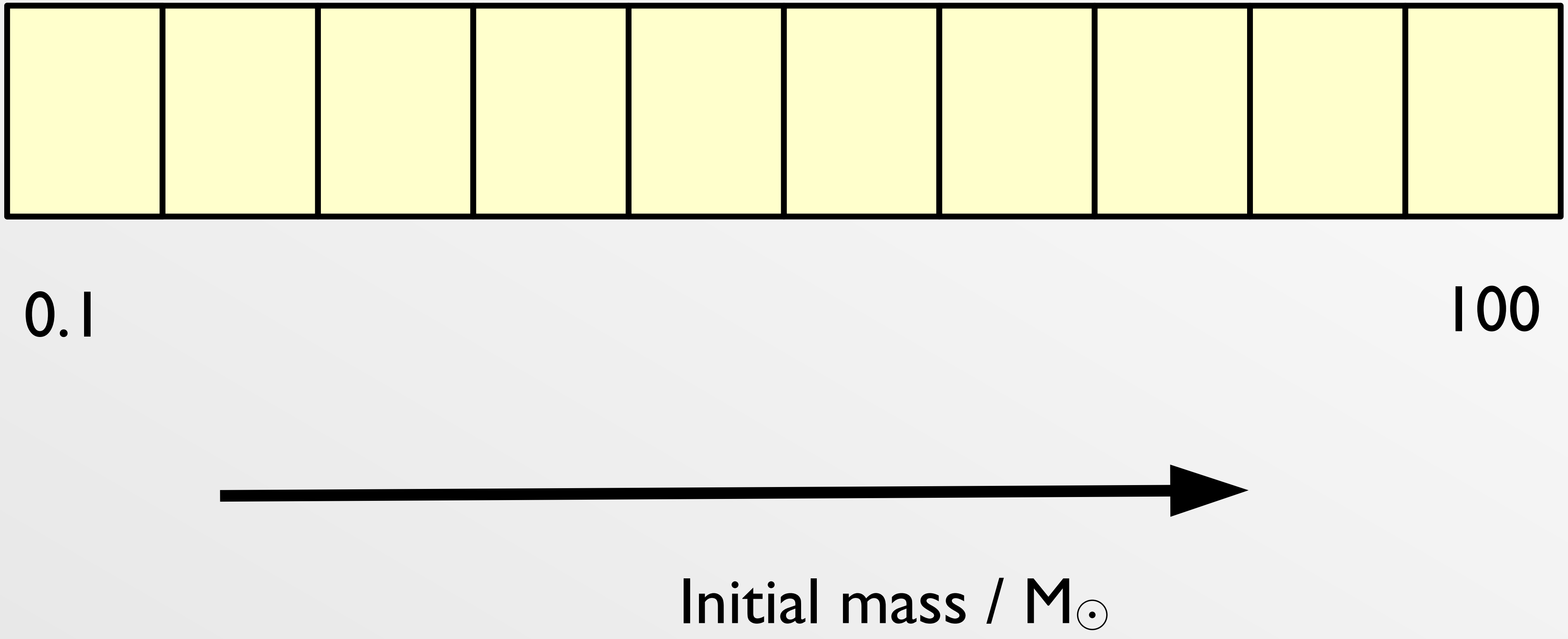
Parameter space

stellar models

0.1 100

Initial mass / $M_\odot$

Parameter space



Parameter space

5	**15**	**25**	**35**	**45**	**55**	**65**	**75**	**85**	**95**

→

Initial mass / $M_\odot$

Parameter space

2.3	**3.7**	**4.3**	**4.8**	**5.1**	**5.4**	**5.6**	**5.8**	**6.0**	**6.1**

$\log_{10} L/L_\odot$

$$L \approx 1.5\left(\frac{M}{M_\odot}\right)^3$$

5	**15**	**25**	**35**	**45**	**55**	**65**	**75**	**85**	**95**

Initial mass / $M_\odot$

Luminosity statistics

2.3	**3.7**	**4.3**	**4.8**	**5.1**	**5.4**	**5.6**	**5.8**	**6.0**	**6.1**

$\log_{10} L/L_\odot$

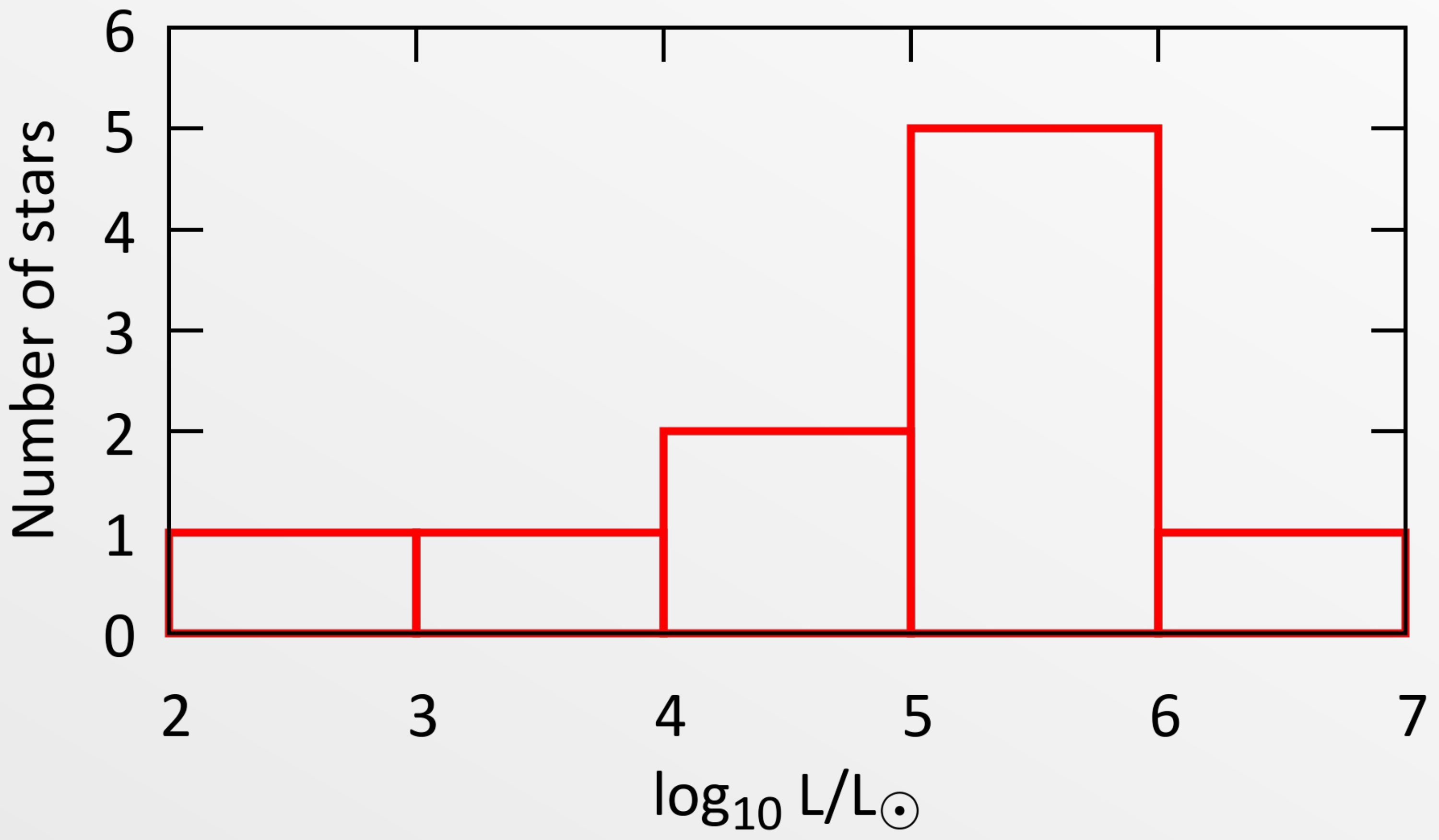
$$\langle \log_{10} L/L_\odot \rangle = 4.92$$

$$\sigma = 1.41$$

Luminosity statistics

2.3	**3.7**	**4.3**	**4.8**	**5.1**	**5.4**	**5.6**	**5.8**	**6.0**	**6.1**

$\log_{10} L/L_\odot$



Does this look like a good model to you?

The need to *P*: Initial distributions

- Stars are not created in equal numbers
- Probability of formation *P* depends on mass

$$\delta P = P(M \text{ to } M + \delta M) = P(M)\,\delta M = \psi(M)\delta M$$

- Or, in binary stars, masses and separation *a*

$$P\left(M_1, M_2, a\right) = \psi\left(M_1\right)\phi\left(M_2\right)\chi\left(a\right)$$

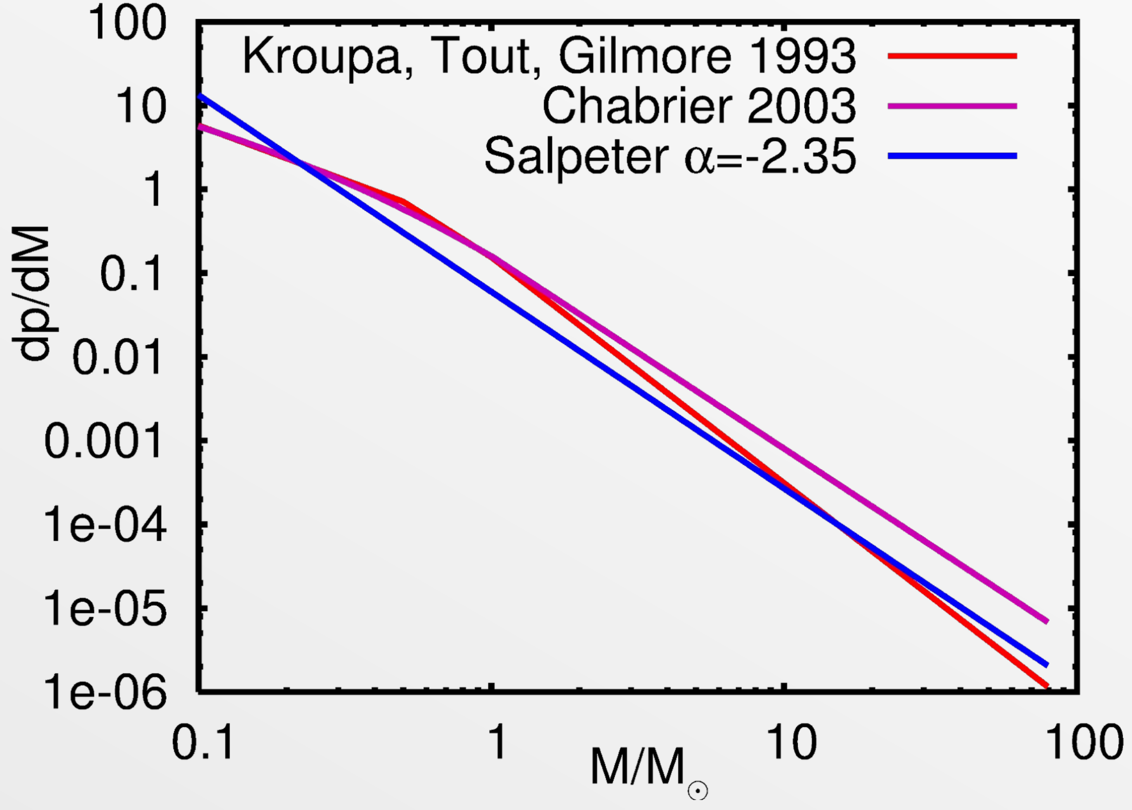
- Function *P* depends on position in parameter space only, not age or stellar evolution.

The need to *P*: Initial distributions

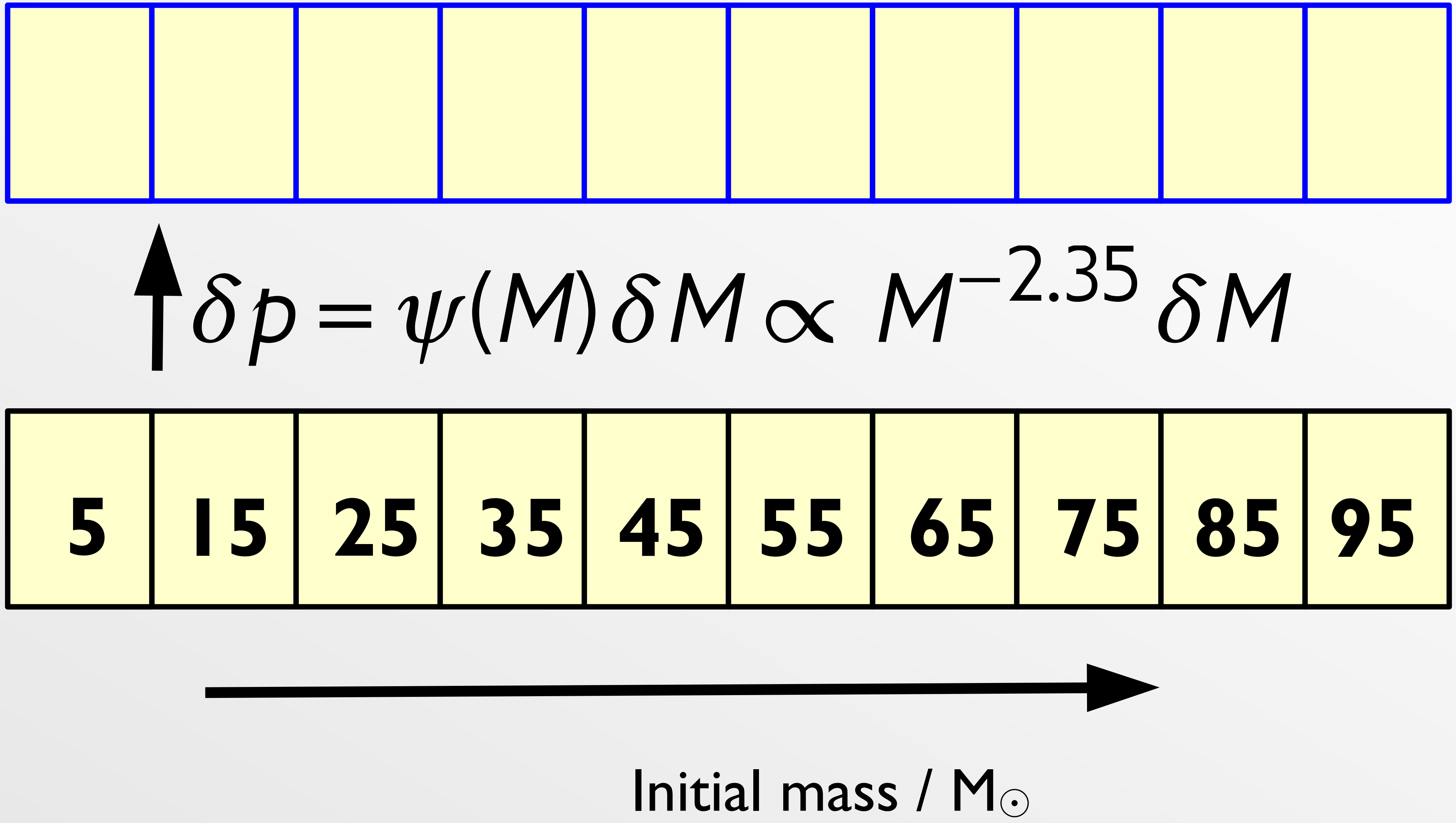
- Stars are not created in equal numbers
- Probability of formation *P* depends on mass, etc.
- e.g. Initial mass function : number of stars by mass

$$dp = \frac{dN}{N_{tot}}$$
$$= \psi(M)dM$$

$$\int \psi(M)dM = 1$$



Initial mass weights, e.g. Salpeter IMF



$$\delta p = \psi(M)\,\delta M \propto M^{-2.35}\,\delta M$$

Initial mass weights, e.g. Salpeter IMF

0.87	0.07	0.02	0.01	0.005	0.004	0.003	0.002	0.001	0.001

δp

$$\delta p = \psi(M)\,\delta M \propto M^{-2.35}\,\delta M$$

5	15	25	35	45	55	65	75	85	95

Initial mass / $M_{\odot}$

Weighted luminosities

0.87	0.07	0.02	0.01	0.005	0.004	0.003	0.002	0.001	0.001

δp

$\times$

2.3	3.7	4.3	4.8	5.1	5.4	5.6	5.8	6.0	6.1

$\log_{10} L/L_\odot$

Weighted luminosities

0.87	0.07	0.02	0.01	0.005	0.004	0.003	0.002	0.001	0.001

×

2.3	3.7	4.3	4.8	5.1	5.4	5.6	5.8	6.0	6.1

=

2.02	0.27	0.10	0.05	0.03	0.02	0.01	0.01	0.008	0.006

Luminosity statistics with IMF

2.02	**0.27**	**0.10**	**0.05**	**0.03**	**0.02**	**0.01**	**0.01**	**0.008**	**0.006**

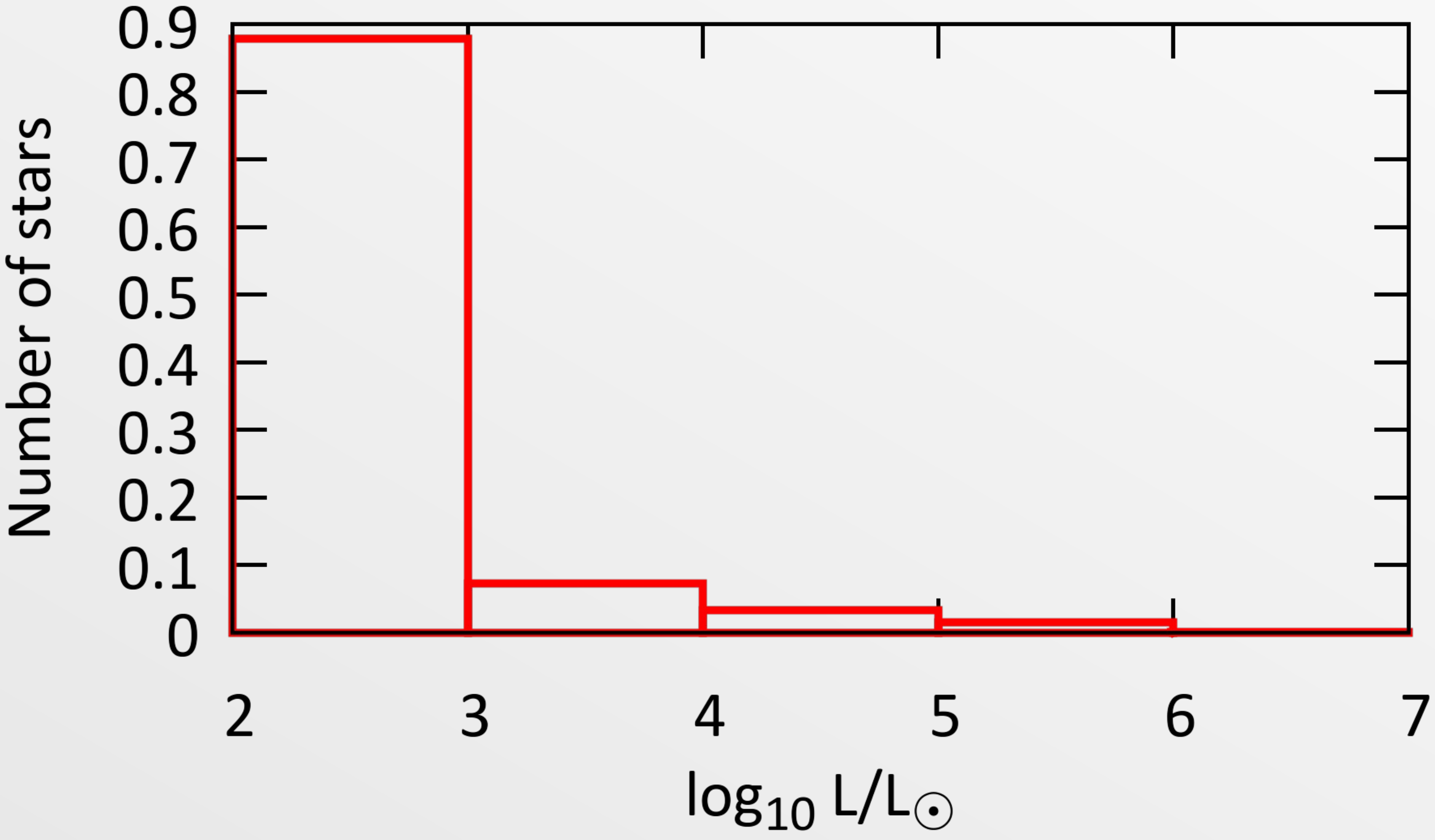
$$\langle \log_{10} L/L_\odot \rangle = 0.25$$

$$\sigma = 0.39$$

Much more like what we see!

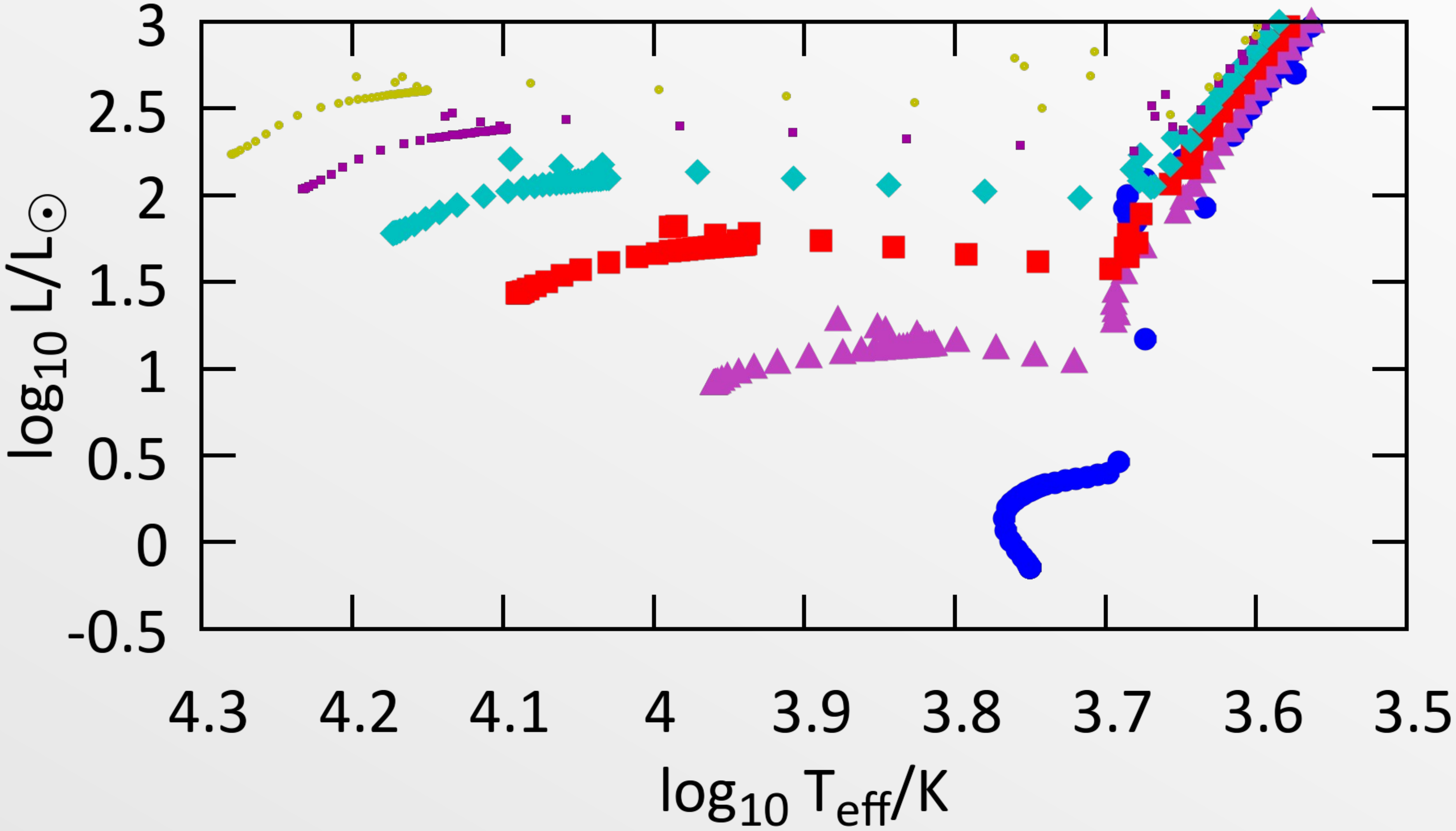
Luminosity statistics with IMF

2.02	0.27	0.10	0.05	0.03	0.02	0.01	0.01	0.008	0.006



Time dependence 1: timesteps

- Discrete time evolution : timesteps δt



Number counts

- Time in a given **evolutionary phase**

$$\Delta t_i = \sum_{t_{min}}^{t_{max}} \bar{\delta}(t)\,\delta t$$

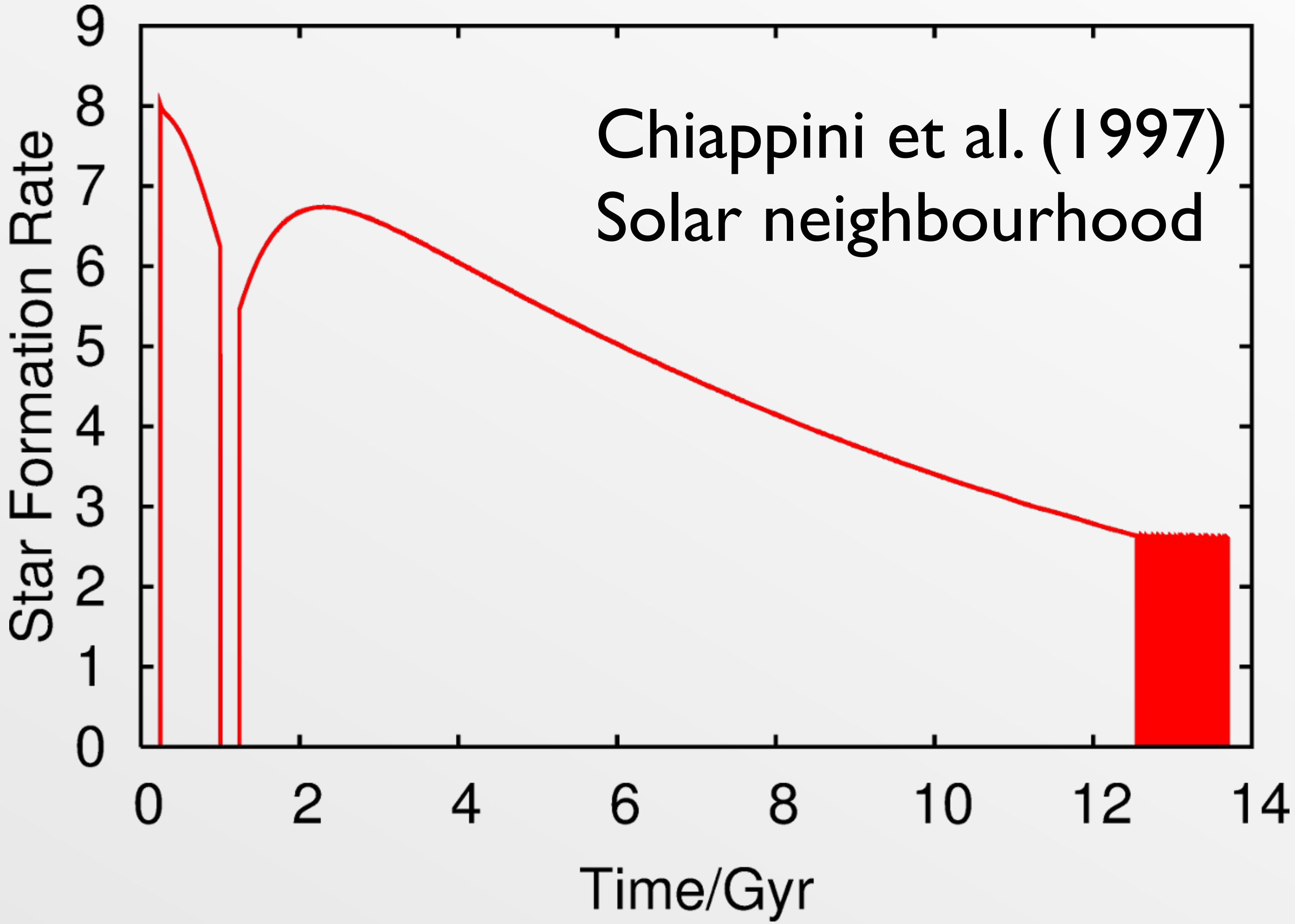
where $\bar{\delta}(t) = 1$ during the phase,

$= 0$ otherwise.

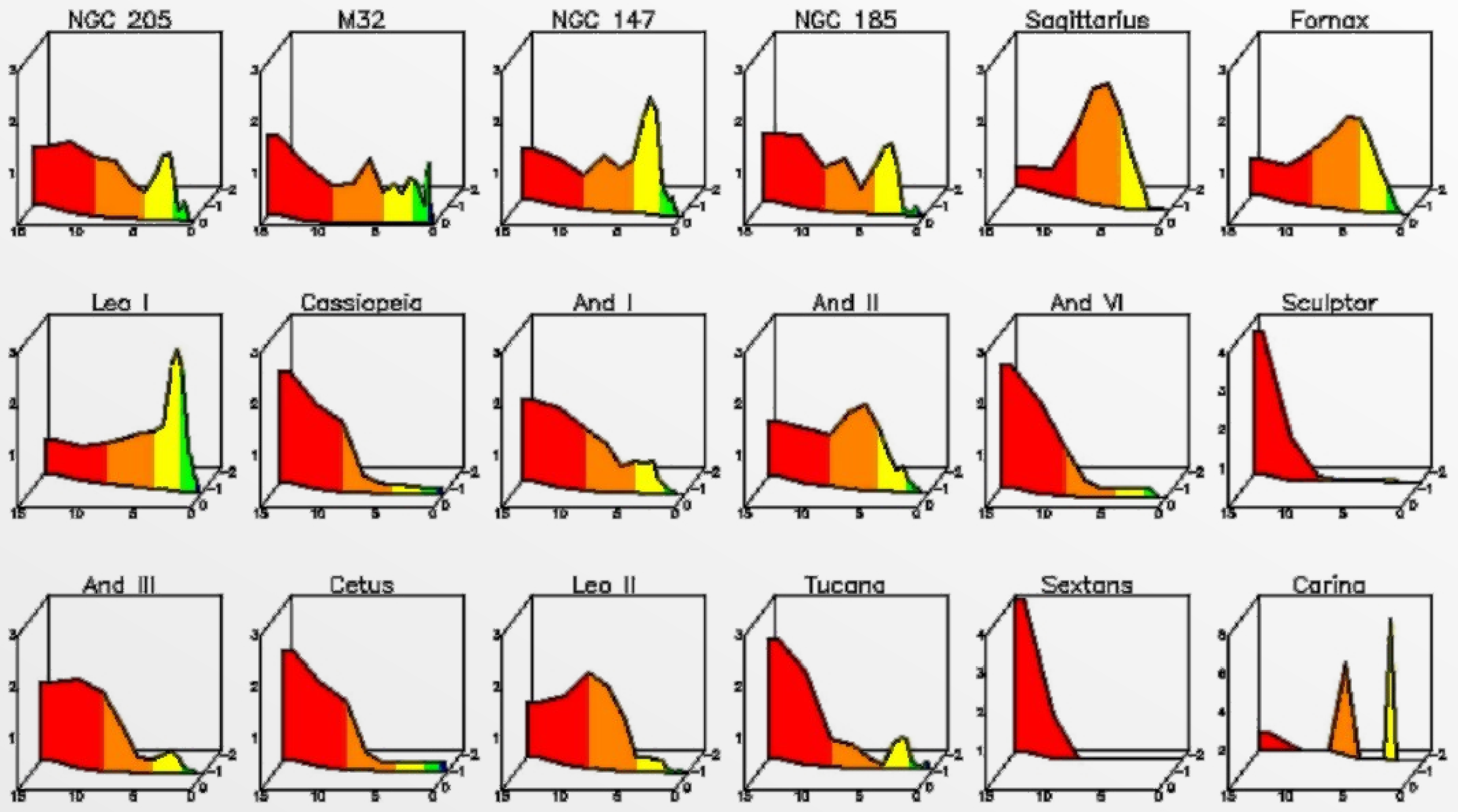
$\bar{\delta}(t)$ contains all the **stellar evolution**

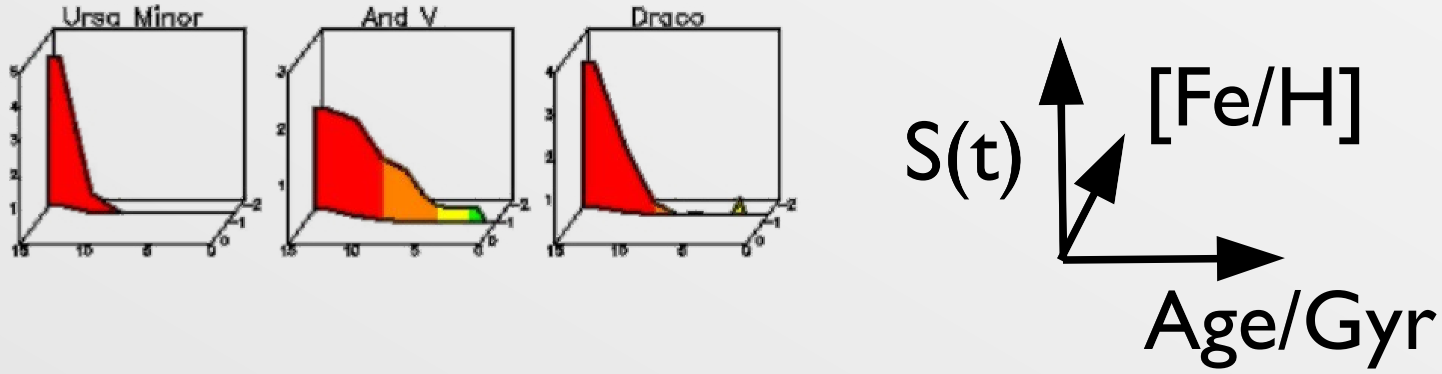
Time dependence 2: star formation

- Star formation rate is a function of time *S(t)*



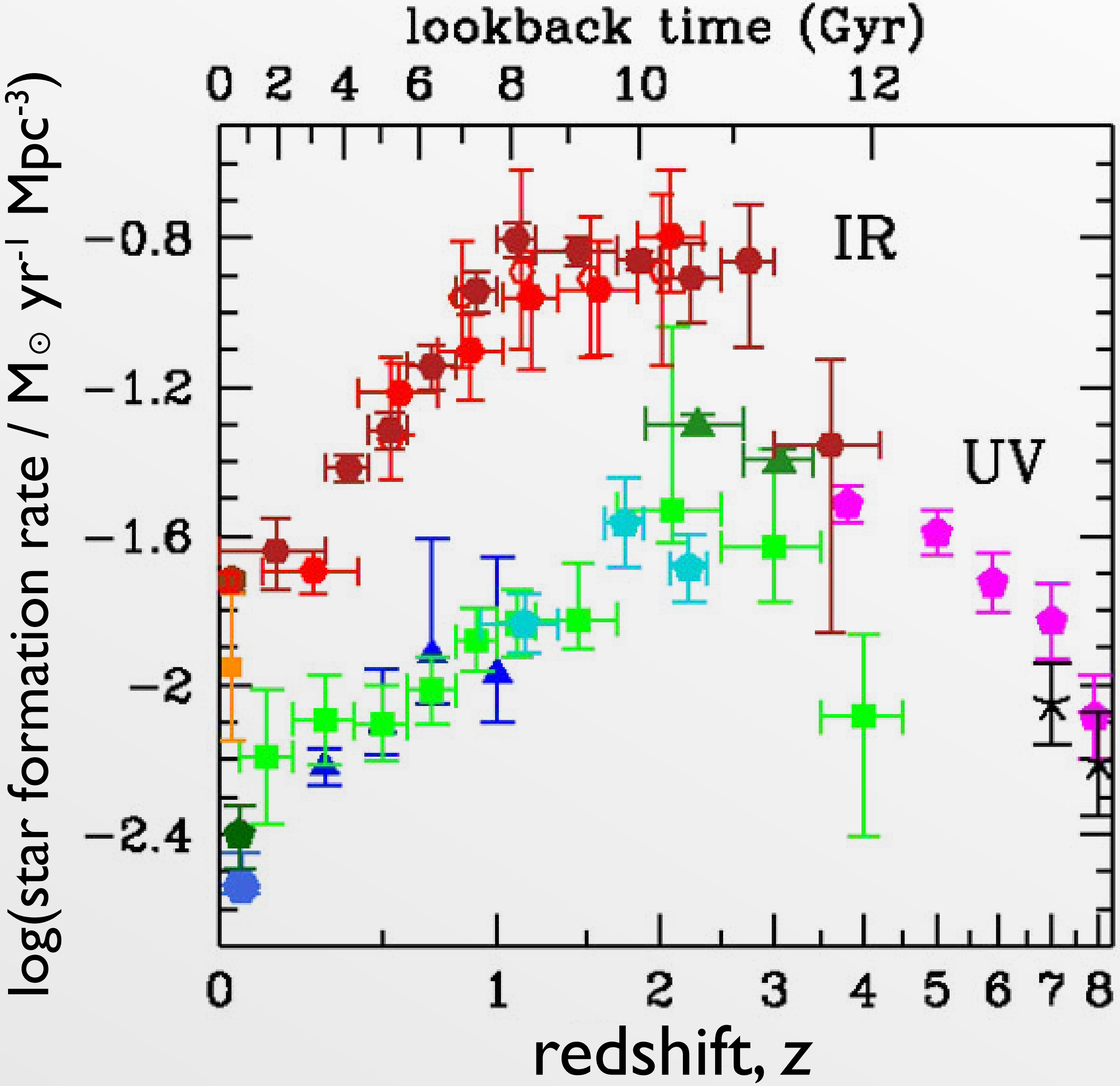
Time dependence 2: star formation





from Dolphin (2005)

Cosmic star formation



Madau & Dickinson
2014 ARAA review

Many stars are multiple

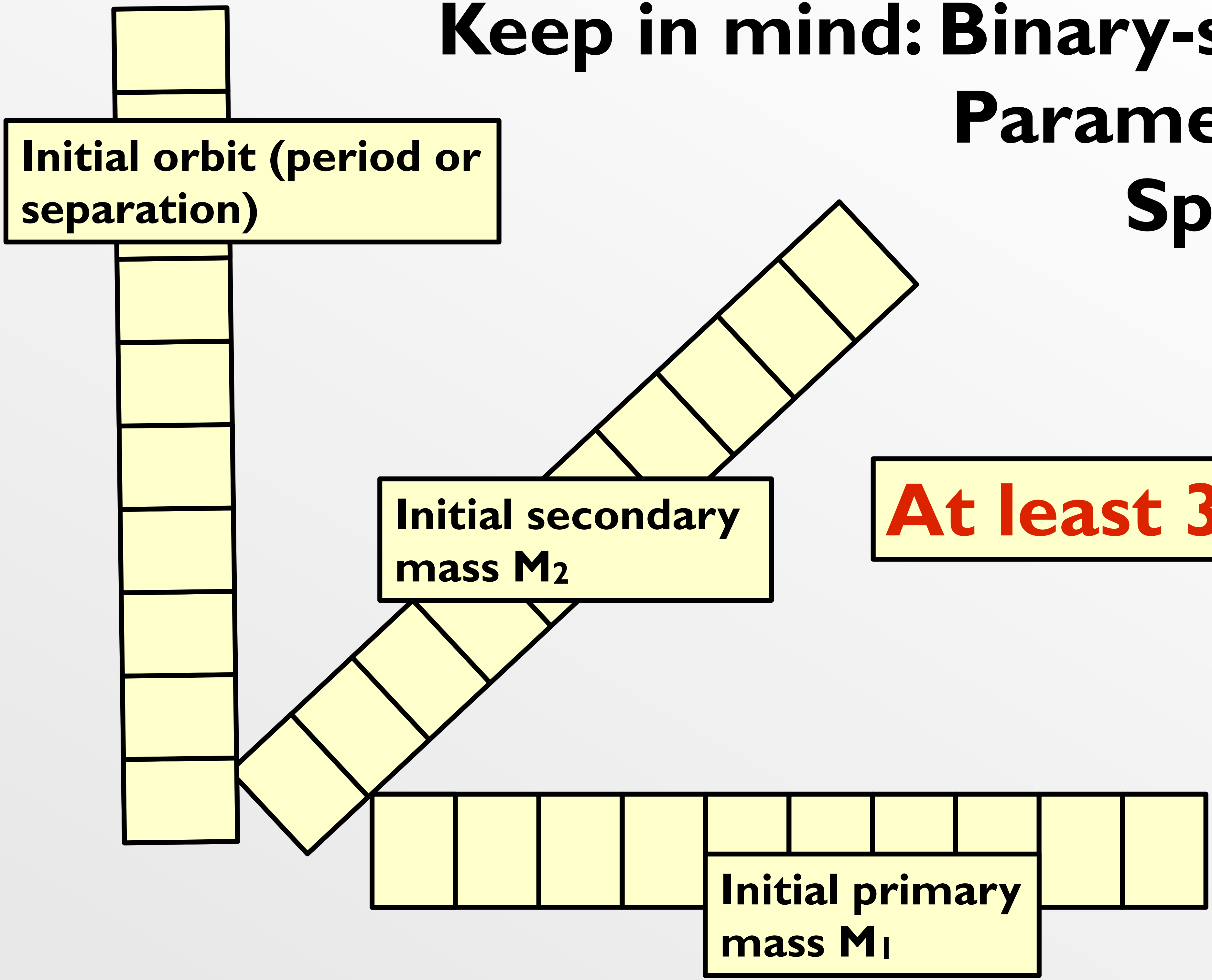
- Binary fraction is ~70% for $M > 8 M_\odot$!

(Sana+ 2012, Moe & Di Stefano)

- Each system now has M_2, period (P) or separation (a), eccentricity (e)
- More model uncertainties e.g.
  - Mass transfer and its efficiency
  - Common envelope evolution
  - Merging, rejuvenation, tides, angular momentum loss
  - Accretion, subsequent mixing, winds, SNe Ia, novae

BACK TO THIS LATER!

Keep in mind: Binary-star Parameter Space



At least 3D!

Computation time

- **Single** stars

  N x 1 hour = N hours = **10 hours** with N=10

- **Binary** stars

  N x N x N x 2 hours = $2N^3$ hours = **2000 hours**

- An expensive problem!
- Especially for rare channels which need $N \sim 100$
- Much more on this later... **but keep it in mind**.

Numbers of stars

For a given star, label *i*, contribution is

$$n_i = S(t) \times \psi_i \times \Delta t_i$$

For a stellar population

$$N = \sum_i \sum_{t_{min}}^{t_{max}} S(t)\,\psi_i\,\bar{\delta}_i(t)\,\delta t_i$$

Expensive double sum! **Convolution** problem

Numbers of stars

For a stellar population of only binaries:

$$N = \sum_{i} \sum_{t_{\min}}^{t_{\max}} S(t)\, \Psi_i(M_1, M_2, a)\, \bar{\delta}_i(t)\, \delta t$$

where *i* is all stars, i.e. in M_1, M_2 and *a* space.

$$\Psi(M_1, M_2, a) = \psi(M_1)\, \phi(M_2)\, \chi(a)$$

Assumes separable function...

Neglects eccentricity e and other parameters.

Simplifications 1: fix Z

- Assume **constant metallicity** Z and other physics

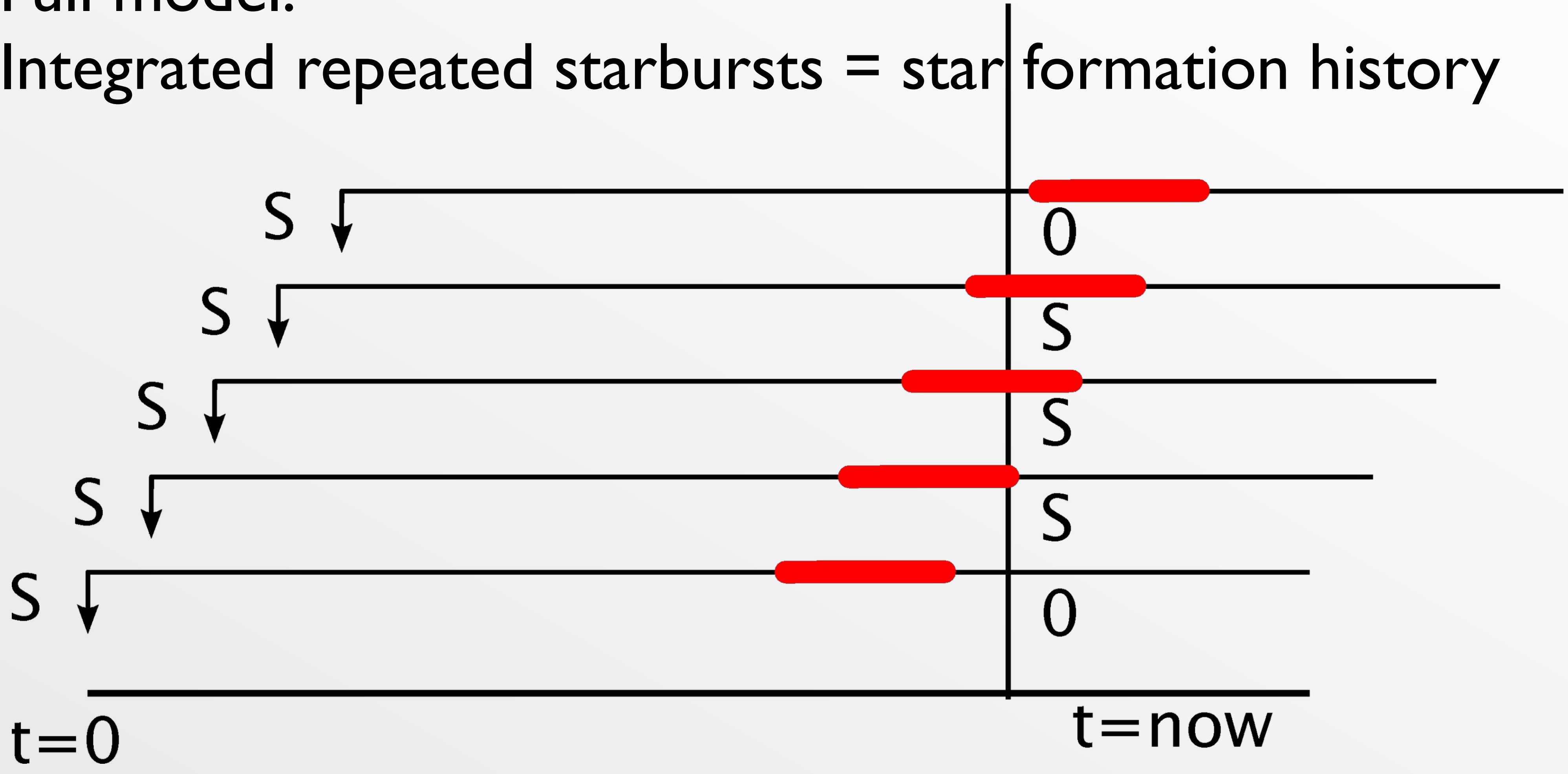
$\bar{\delta}_i(t)$ is a function of metallicity Z

If we fix the metallicity Z we need only one set of stellar evolution models: this is much easier and faster!

We can still vary *S*(*t*) cheaply

Full model:

Integrated repeated starbursts = star formation history

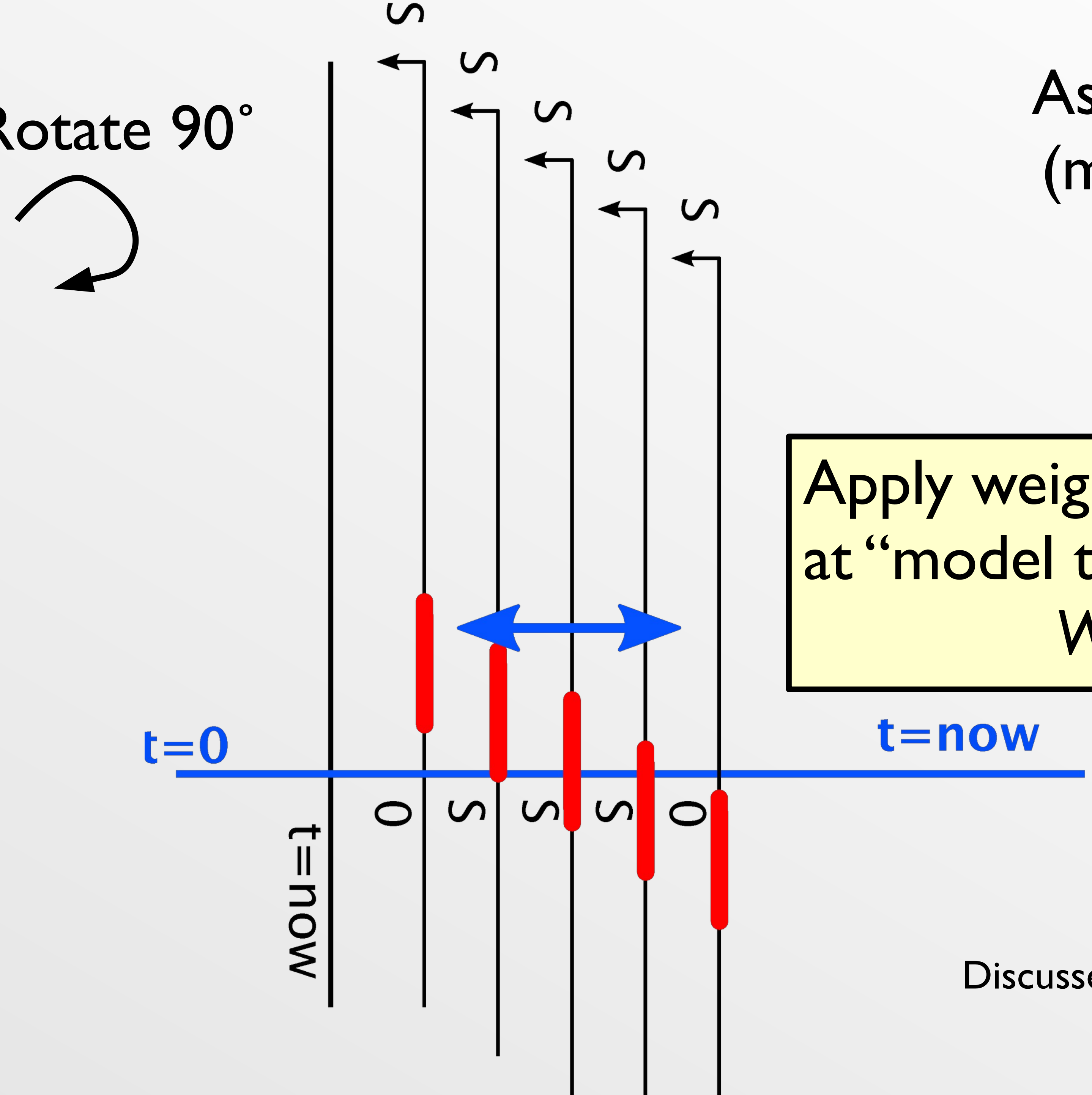


Red = phase of interest (e.g. red giant branch)

Intersection with "now" → stellar number counts

Variable *S(t)* from a single starburst!

Rotate 90°



Assuming physics (metallicity, etc.) constant

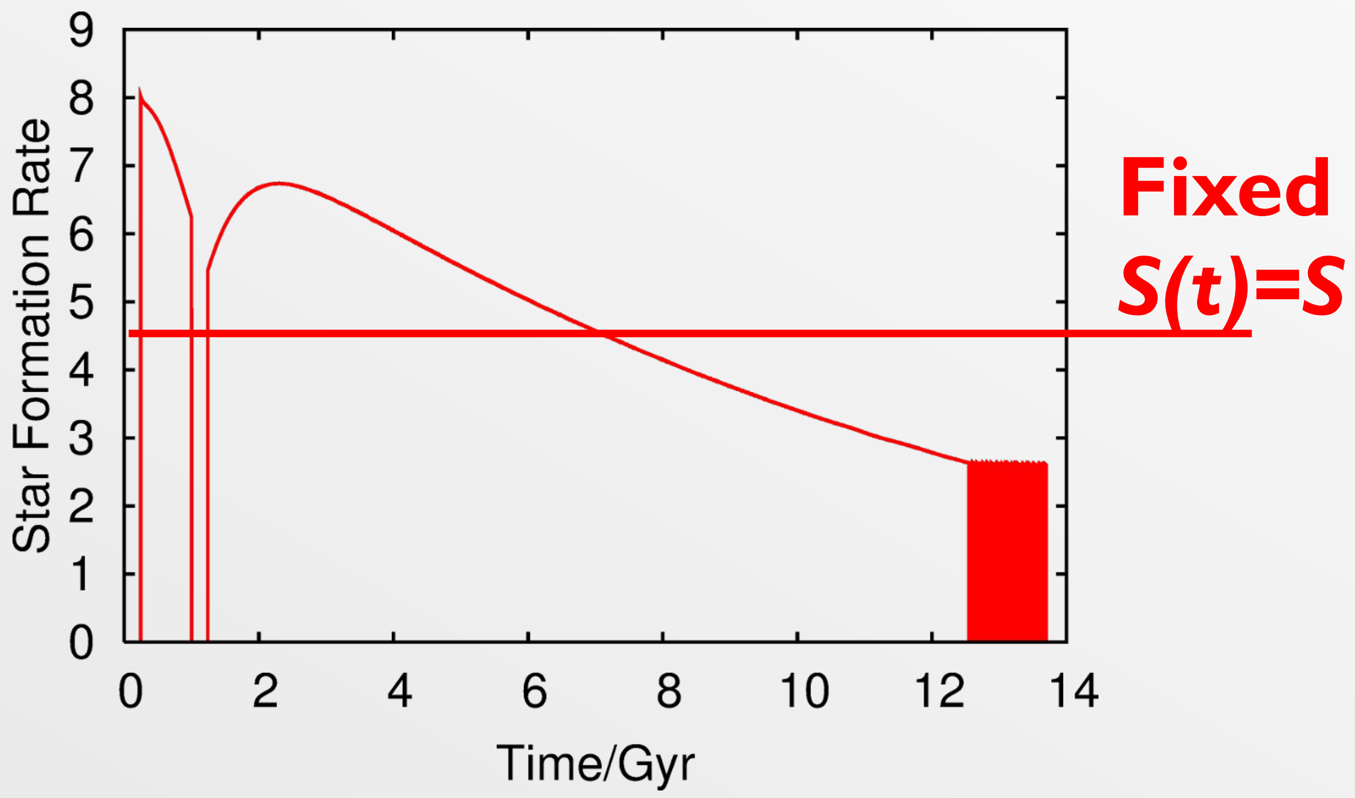
Apply weighting function at "model time" *t*':

$W(t') = S(\text{now}-t)$

Discussed in Izzard & Halabi (2018)

Simplifications 2: fix *SFR*=*S*(*t*)

- Assume constant star formation rate



If we calculate number ratios $\tilde{N}$ *and* relative rates $\tilde{R}$ then S cancels!

Equivalent to S=1

True rate is ~ this uncertain anyway!

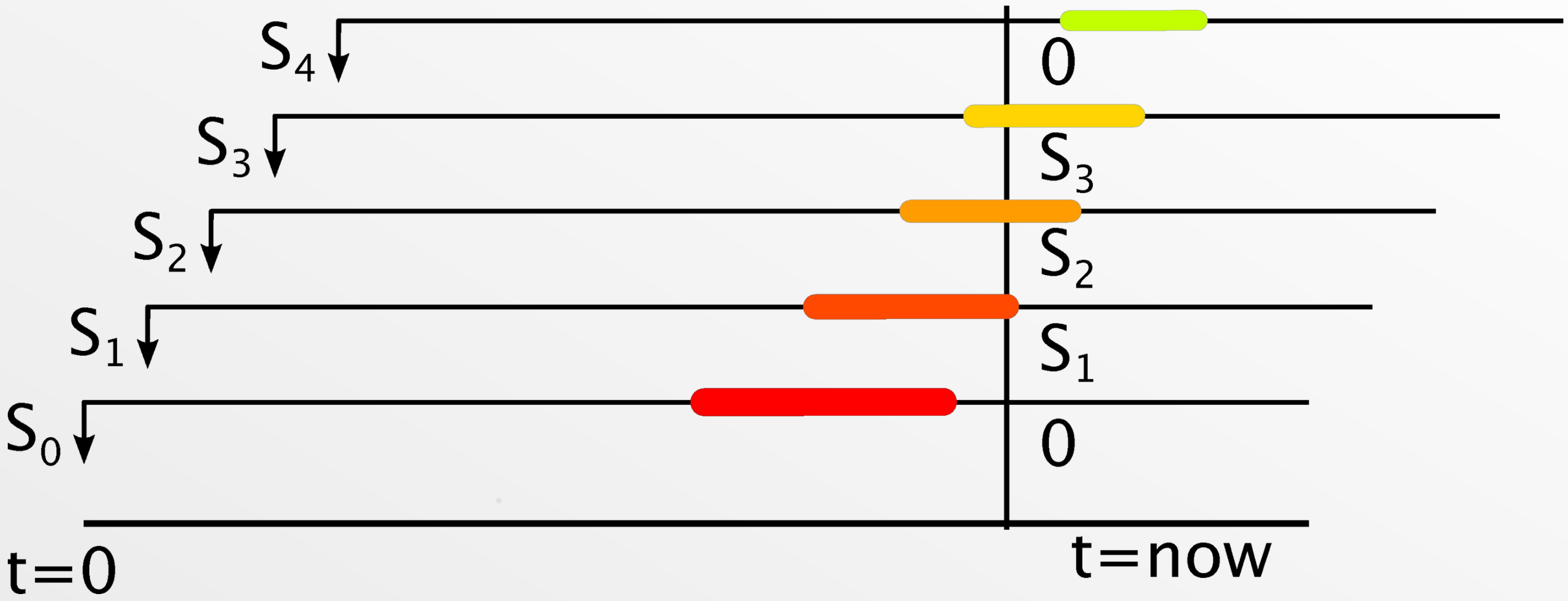
Relative **number** and **rate** counts

$$\tilde{N} = \sum_{\text{all stars}} \sum_{\text{all timesteps}} \left(\bar{\delta} \times \delta t \times \delta P\right)$$

$$\tilde{R} = \sum_{\text{all stars}} \sum_{\text{all timesteps}} \left(\bar{\delta} \times \delta P\right)$$

***S* cancels out : uncertainties in *S* cancel too!**
This is the calculation performed in many population synthesis studies.
Often it really is "good enough".

The ultimate ideal model



- Vary *Z(t)*, *S(t)*
- Include binary stars
- Stellar interactions?
- This is full *N*-body
  Galactic Chemical Evolution

Recap:

- Population synthesis = a **combination** of many stellar **models** to make **statistical** predictions
- Compare these to "**reality**"
- Like being an accountant **without the salary** (sorry!)
- But you get to keep your **soul :)**
- Lots of adding up to do ...

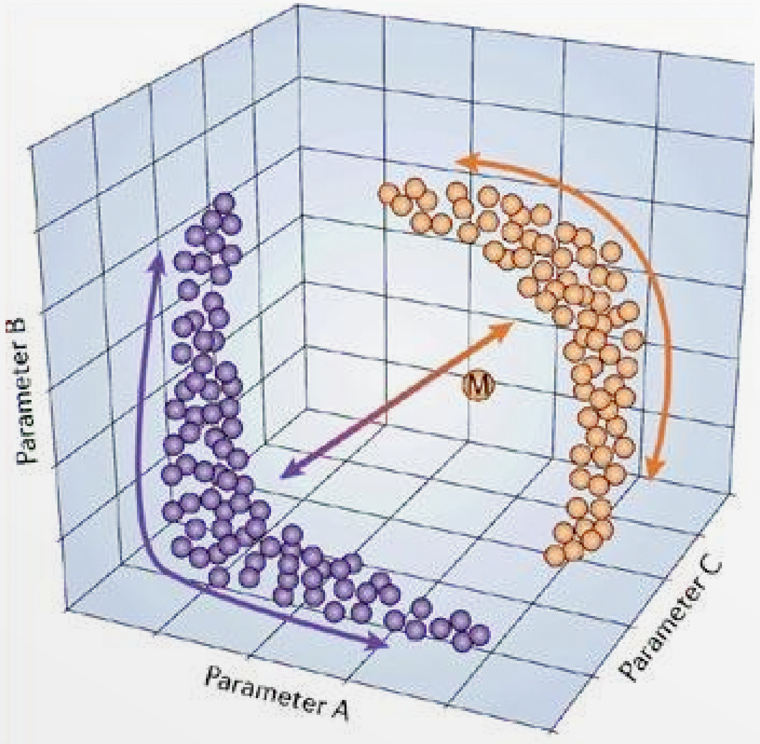
Parameter space grids and rapid stellar models

- Our uniform mass grid was **stupid**: can do better
- What about stellar evolution models?

  What are used in pop syn?

  How do we incorporate them?

  Are they **fast enough**?



Parameter space missions

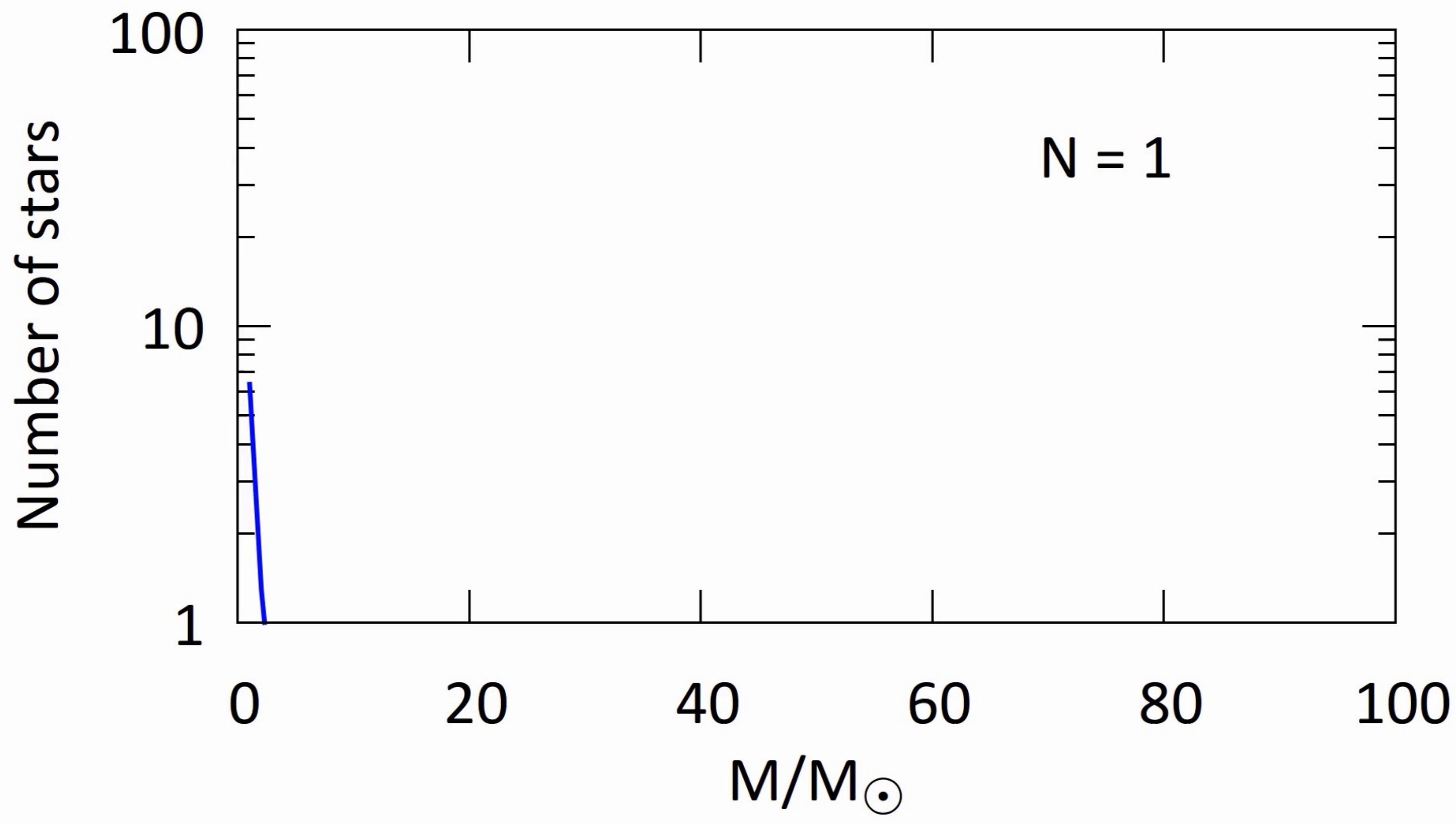
Our parameter space is *n*-dimensional, at least:

- $n = 1$ for **single** stars (mass)
- $n = 3$ for **binary** stars (primary, secondary, separation)
- Can be more! e.g. metallicity, eccentricity, etc.
- Computation time at least ~ N or $2N^3$ for resolution N

What is the best way to sample the (huge!) parameter space?

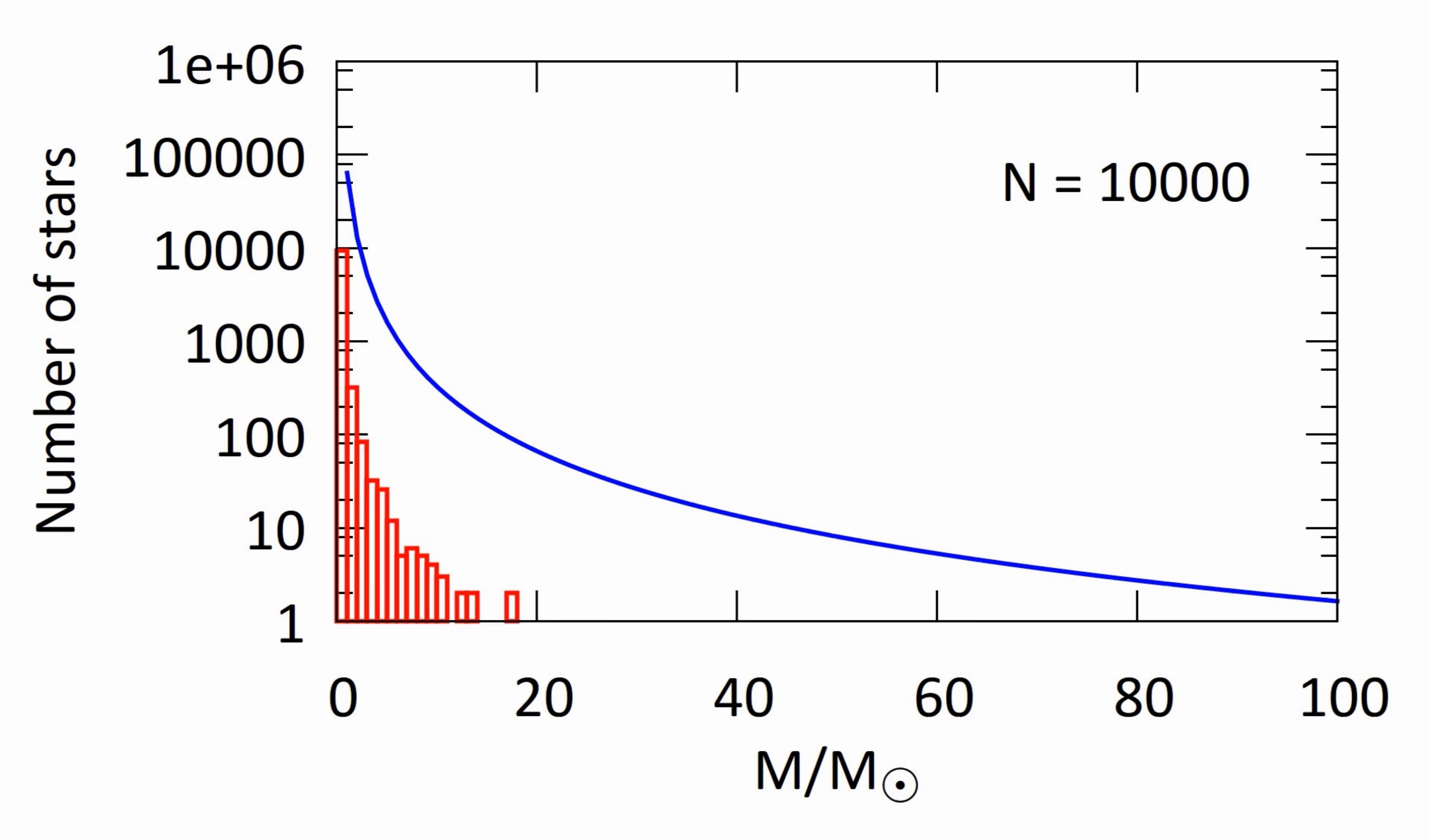
Monte Carlo approach

- Make an initial population using random numbers
- e.g. Salpeter distribution with *N*=100



Monte Carlo approach

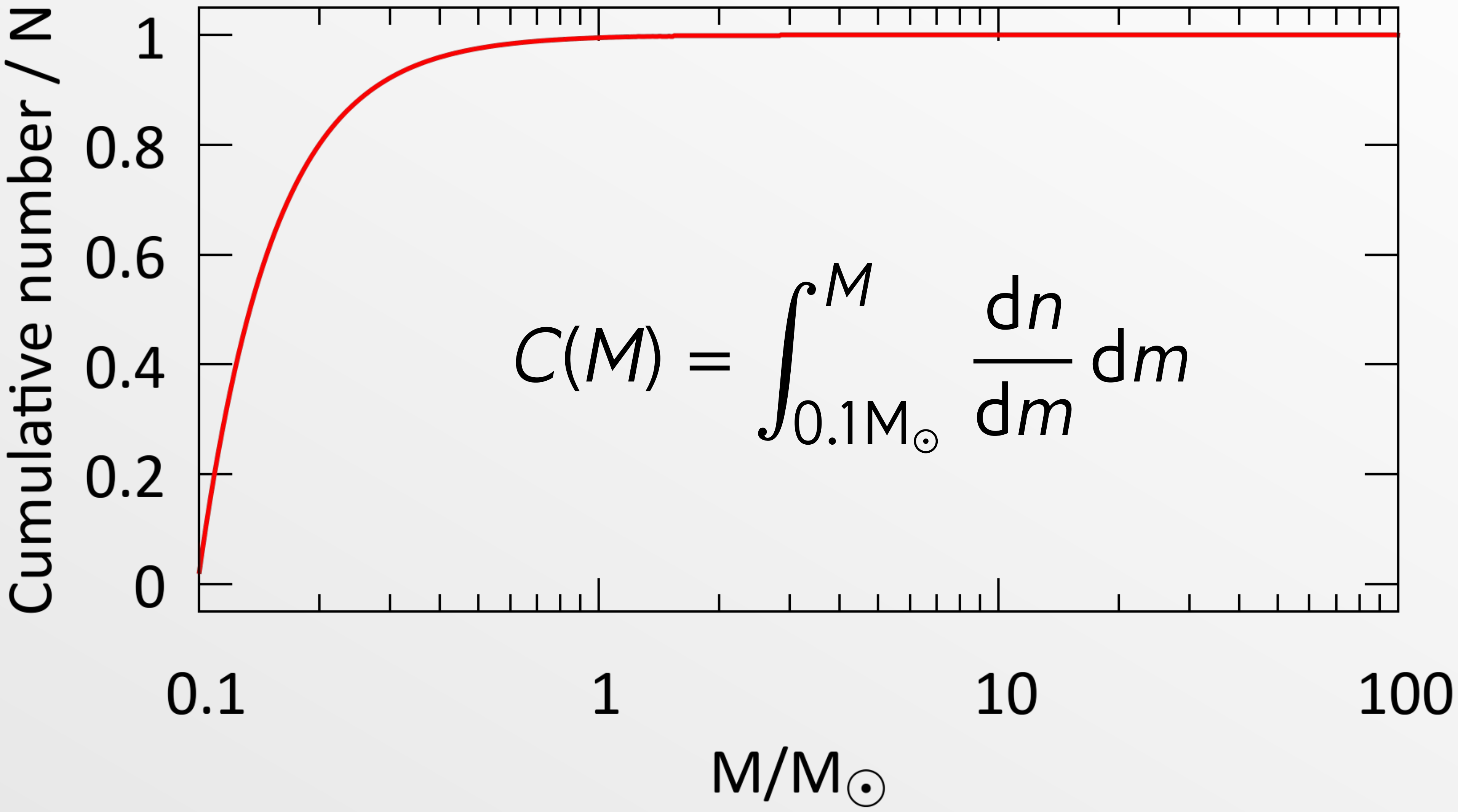
- Make an initial population using random numbers
- e.g. Salpeter distribution with $N=10^4 - 10^6$



Exercise: Monte Carlo algorithm

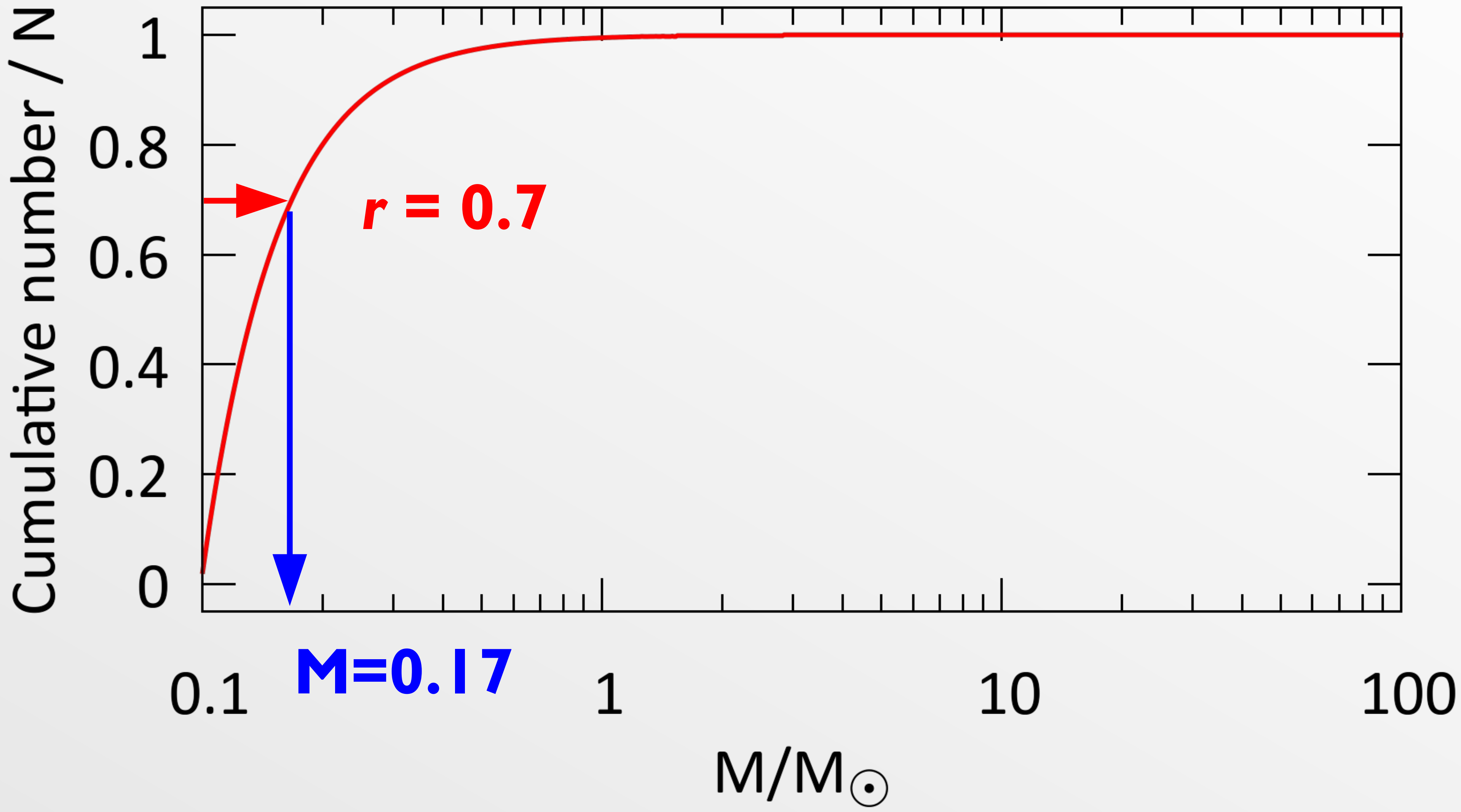
- Map a **random number** $0 < r < 1$ to a mass M
- Salpeter distribution: $$\frac{dn}{dM} = AM^{-2.3}$$
- Normalize: $$\int_{0.1\,M_\odot}^{100\,M_\odot} AM^{-2.3}\,dM = N$$
- The map can be done algebraically, but...
- More flexible to do it numerically, i.e. for ***any function***

- First, calculate a **cumulative distribution function C(*M*)**
  I did it numerically, but in this case the integral is simple.



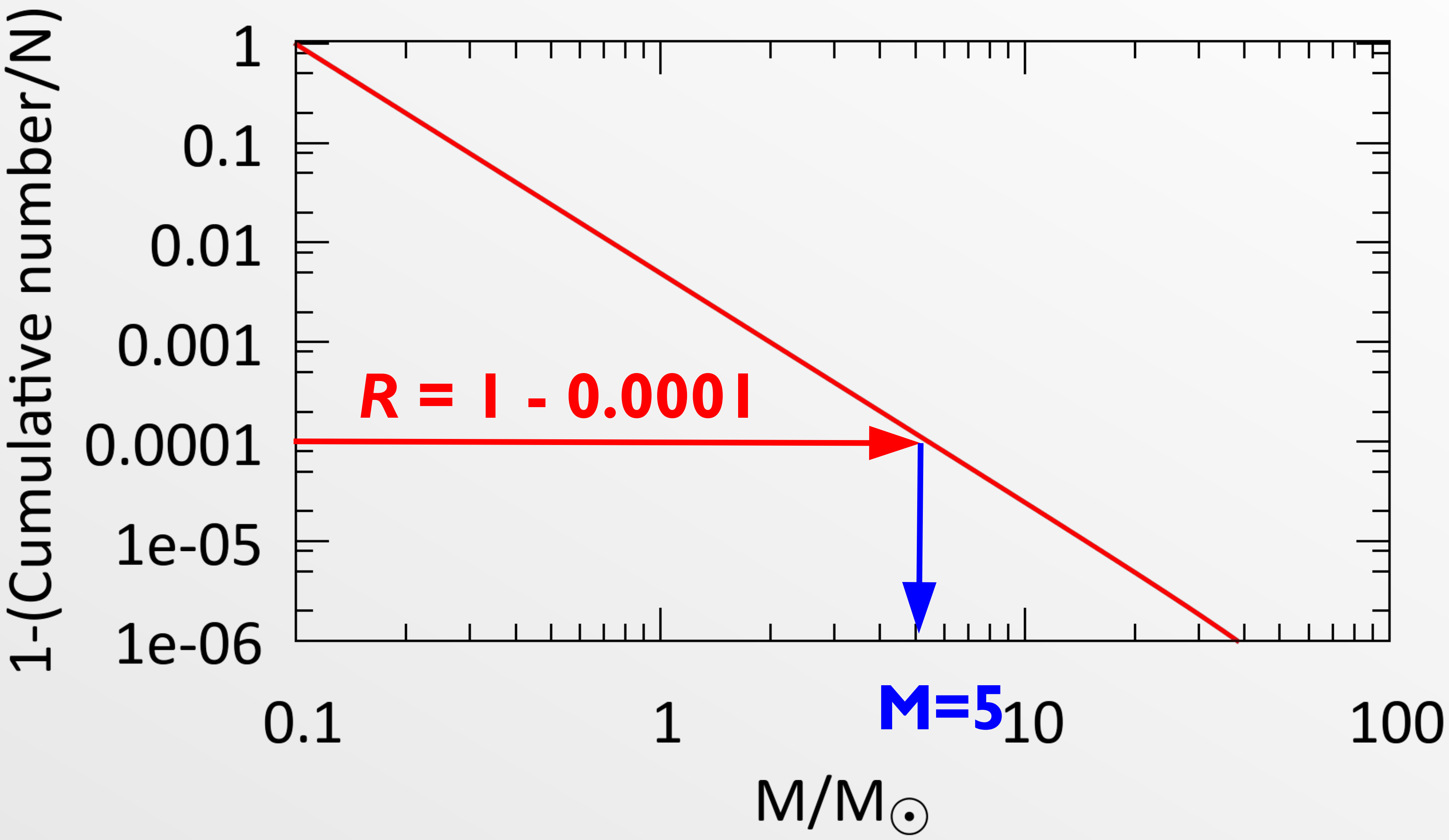
$$C(M) = \int_{0.1M_\odot}^{M} \frac{dn}{dm}\,dm$$

- Second, choose a random number $0 < r < 1$
- This is your **y *axis*** value: **map** it to a mass *M* on the *x-axis*



I do the map using a simple, fast **linear interpolation** code *librinterpolate*

- Example with a different number (on a log plot)



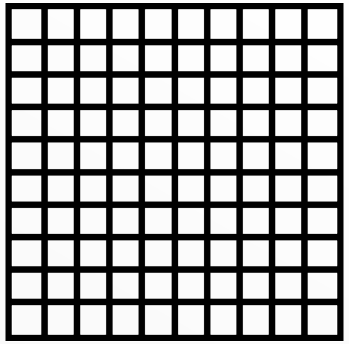
I do the map using a simple, fast **linear interpolation** code *librinterpolate*

Monte Carlo Approach

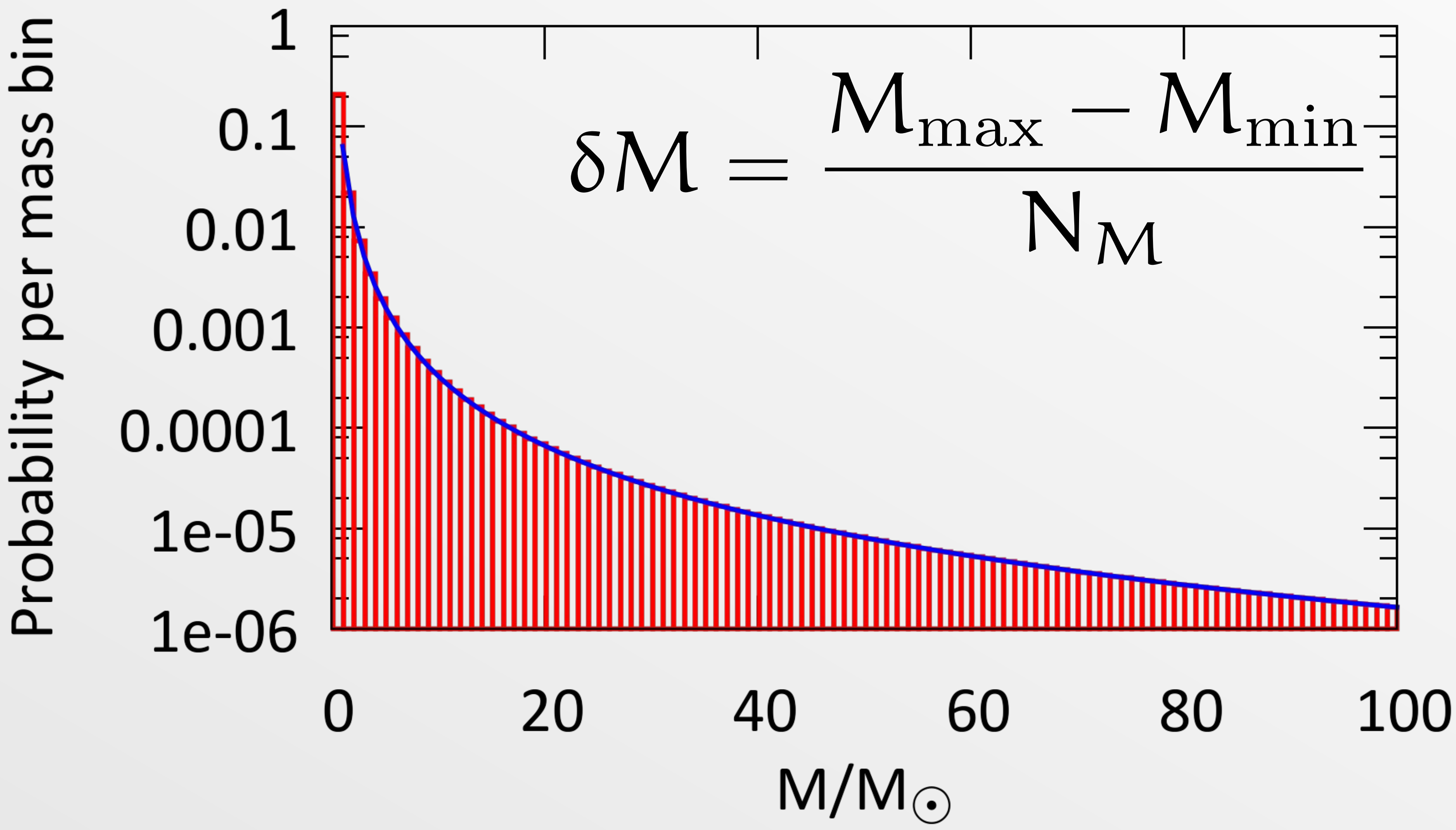
- **Advantages**
  - Simple to implement
  - More CPU = more stars *N* = more resolution
  - Like a "real" survey of stars
  - Repeat: models natural fluctuation.
- **Disadvantages**
  - Global sampling not guaranteed if *N* is not (very?) large
  - Stochastic fluctuation in your sample: what you want?
  - Hard to increase resolution where required (e.g. high M)
  - Rerun: different result. Harder to test for small *N.*

    cf. statistical "bootstrapping"

Grid approach



- Split parameter space into "boxes"
- Weight each box appropriately



A note on grid spacing

$$\delta M = \frac{M_{\max} - M_{\min}}{N_M}$$

$$\sum_{\text{all stars}} \psi(M)\,\delta M = 1.0$$

N	Sum
100	0.2518
1000	0.9063
10000	0.9988
100000	0.9999

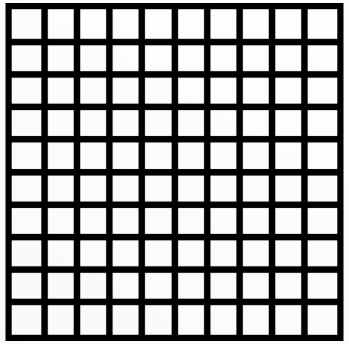
A note on *logarithmic* grid spacing

$$\delta \ln M = \frac{\ln M_{\max} - \ln M_{\min}}{N_M}$$

$$\sum_{\text{all stars}} \psi(M)\, M\, \delta \ln M = 1.0$$

N	Sum
100	0.9997
1000	1.0000
10000	1.0000
100000	1.0000

Grid Approach



- **Advantages**
  - Guaranteed resolution
  - No statistical fluctuation per run
  - Always get the same result for given *N*
  - You choose how to space the grid cells (see isochrones next)
- **Disadvantages**
  - Need to set up (complex?) multi-dimensional grid code
  - Need to calculate weighting functions
  - Not like a “real” survey: like a “perfect” survey

This is the approach I usually use, simple and “good enough”
Can use hybrid grid-MC: e.g. random point in each box

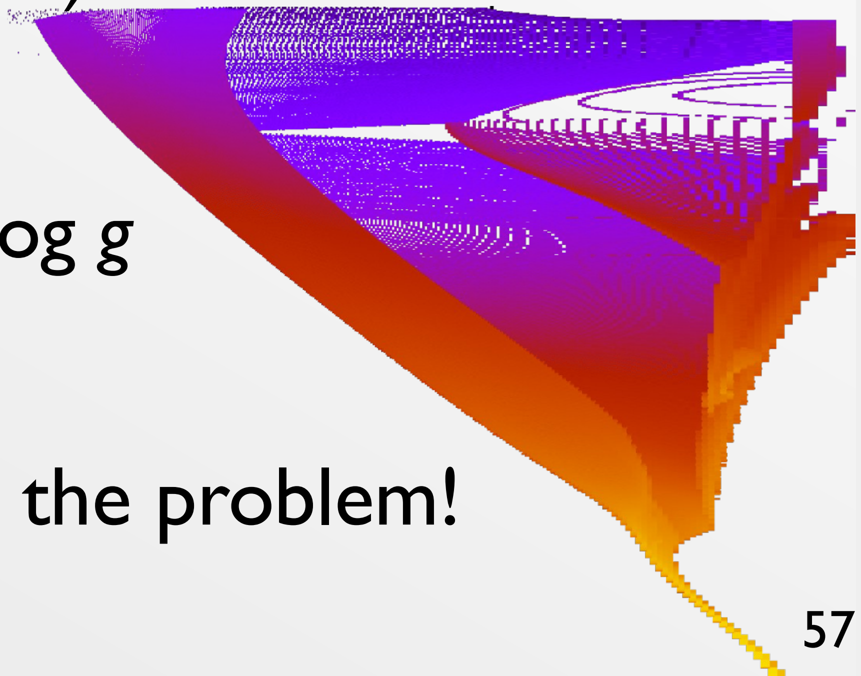
Case study: "3D isochrones"

- I was asked for **isochrones**:

  → stellar properties at a fixed time after a starburst
- Wants 1Myr resolution from 0 to 15Gyr : (N=15000)
- Wants luminosity, temperature, gravity, at high resolution (every 0.05dex or better!)
- This is a lot of data!

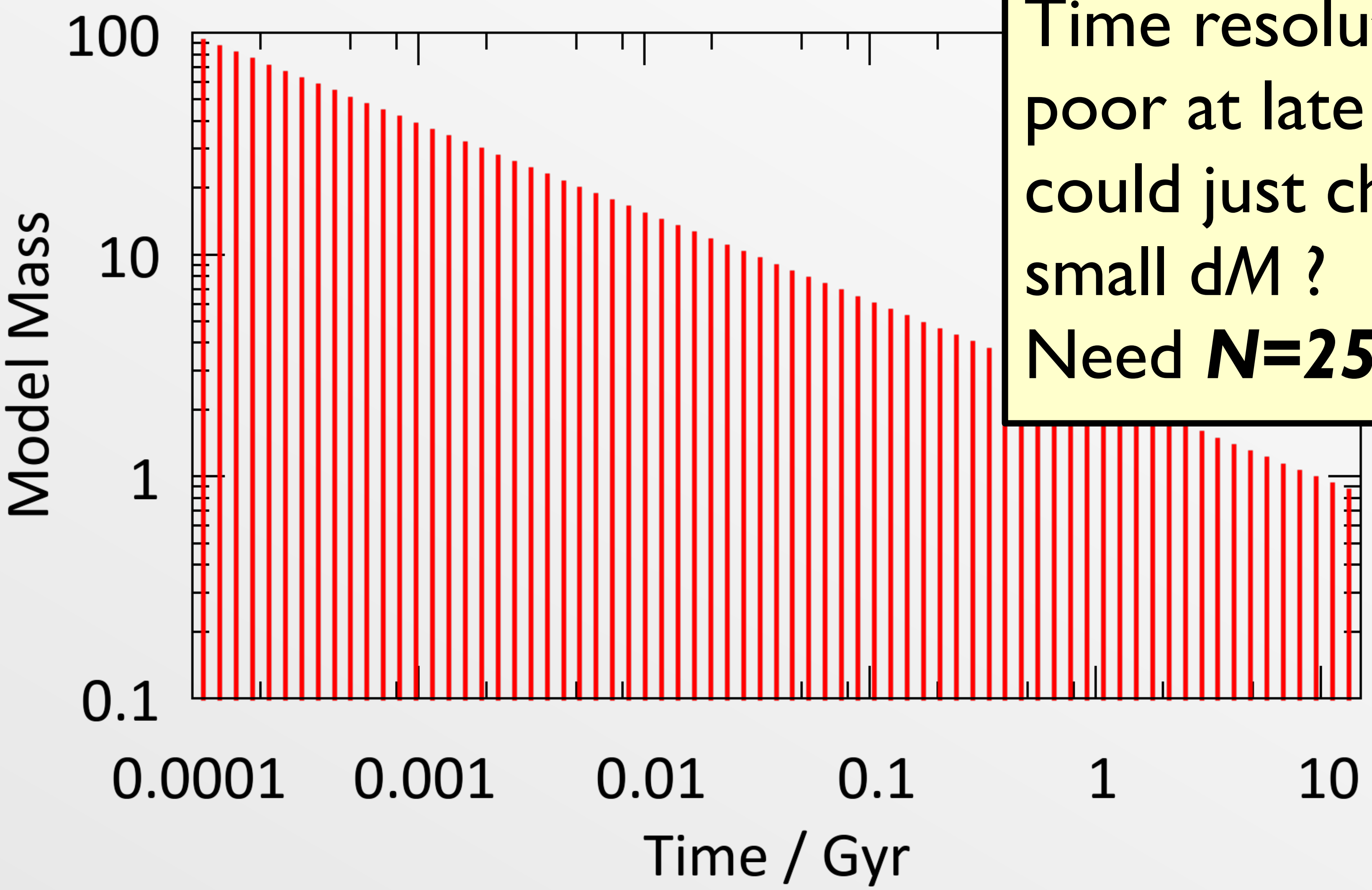
  4D Hypercube of t, log L, log T_{eff}, log g
- What is the best strategy?

  → think before you throw CPU at the problem!



Standard grid

- Choose M > 0.8M$_\odot$, use log mass grid



Time resolution is poor at late times: could just choose small d*M* ?
Need ***N*=250000**

Adaptive grid spacing

- Log L, log T_{eff}, log g are independent of mass M
- But t and M are closely related: at every time t we want to sample at least one star (ideally a few).
- Stellar lifetimes: $t = AM^{x}$ $\quad x \sim -2.5$

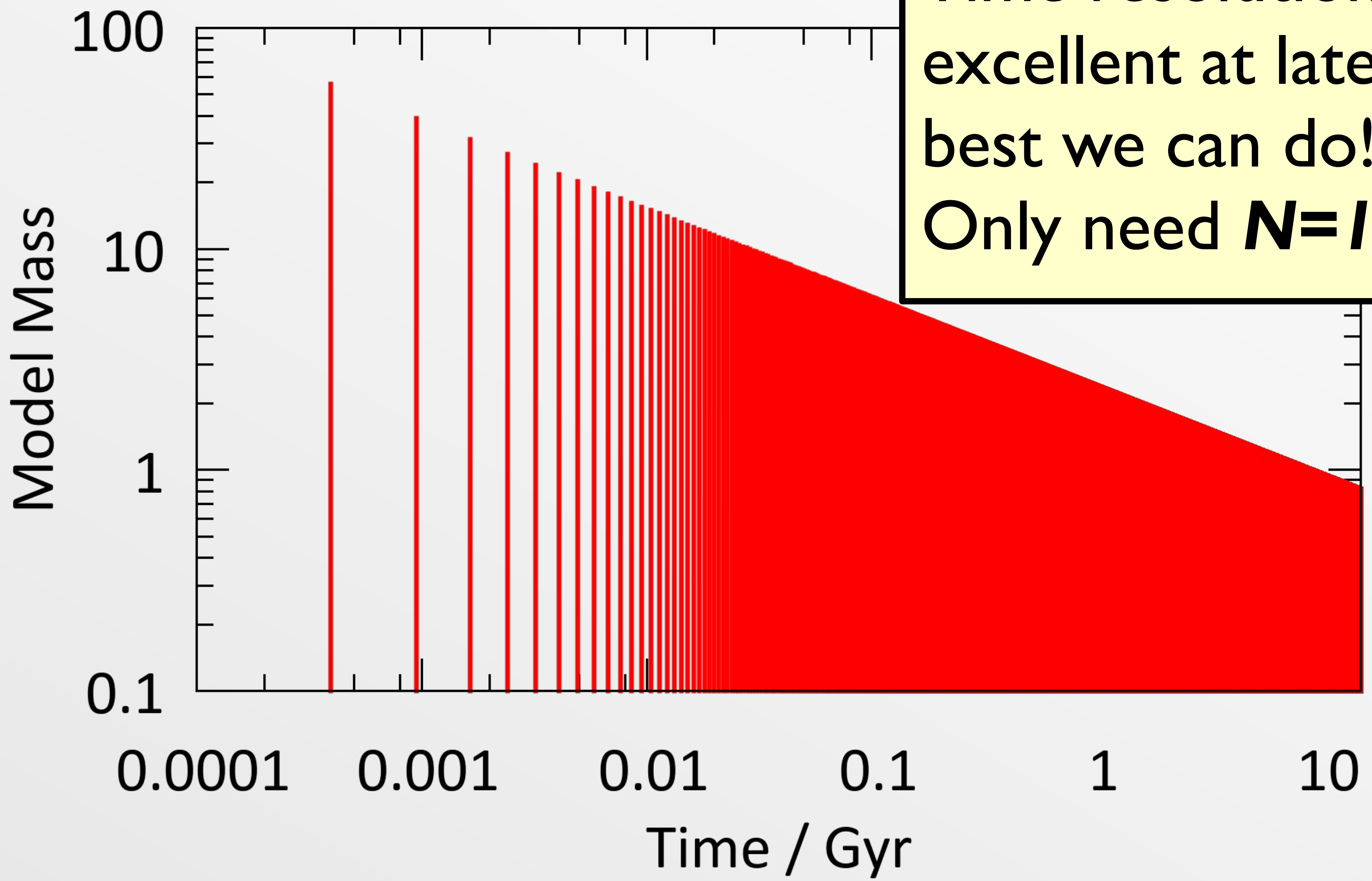
$$\delta t = Ax\,M^{x-1}\delta M$$

$$\delta M = \frac{f\,\delta t}{Ax\,M^{x-1}}$$

Use this as our grid spacing with $f \lesssim 1$

Adaptive grid

- Again choose $M > 0.8 M_{\odot}$

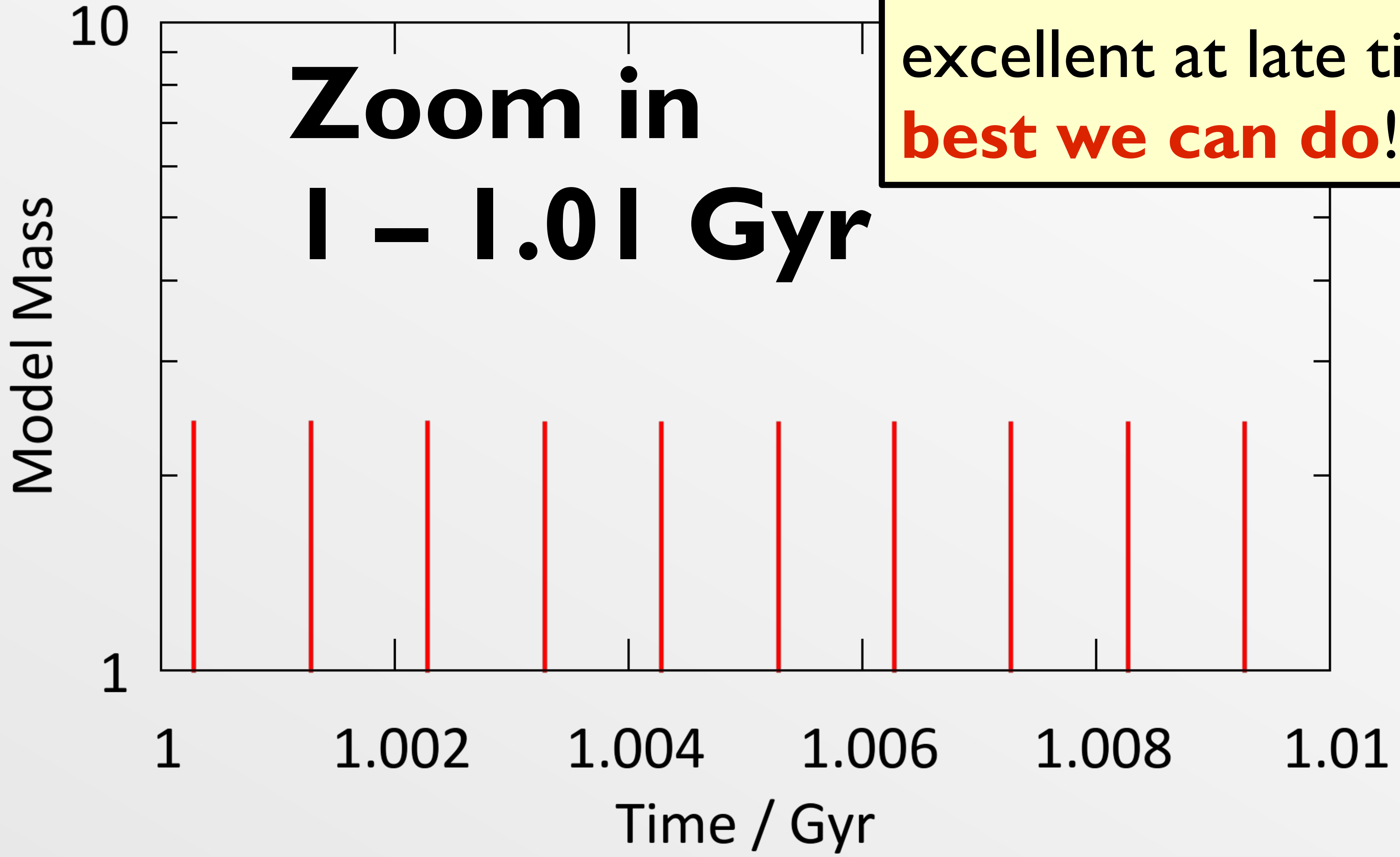


Time resolution is excellent at late times: best we can do!
Only need ***N=16000***

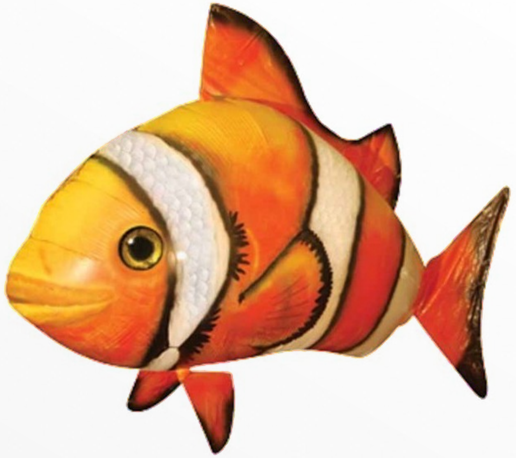
Adaptive grid

- Choose $M>0.8M_\odot$

Time resolution is excellent at late times: **best we can do**!



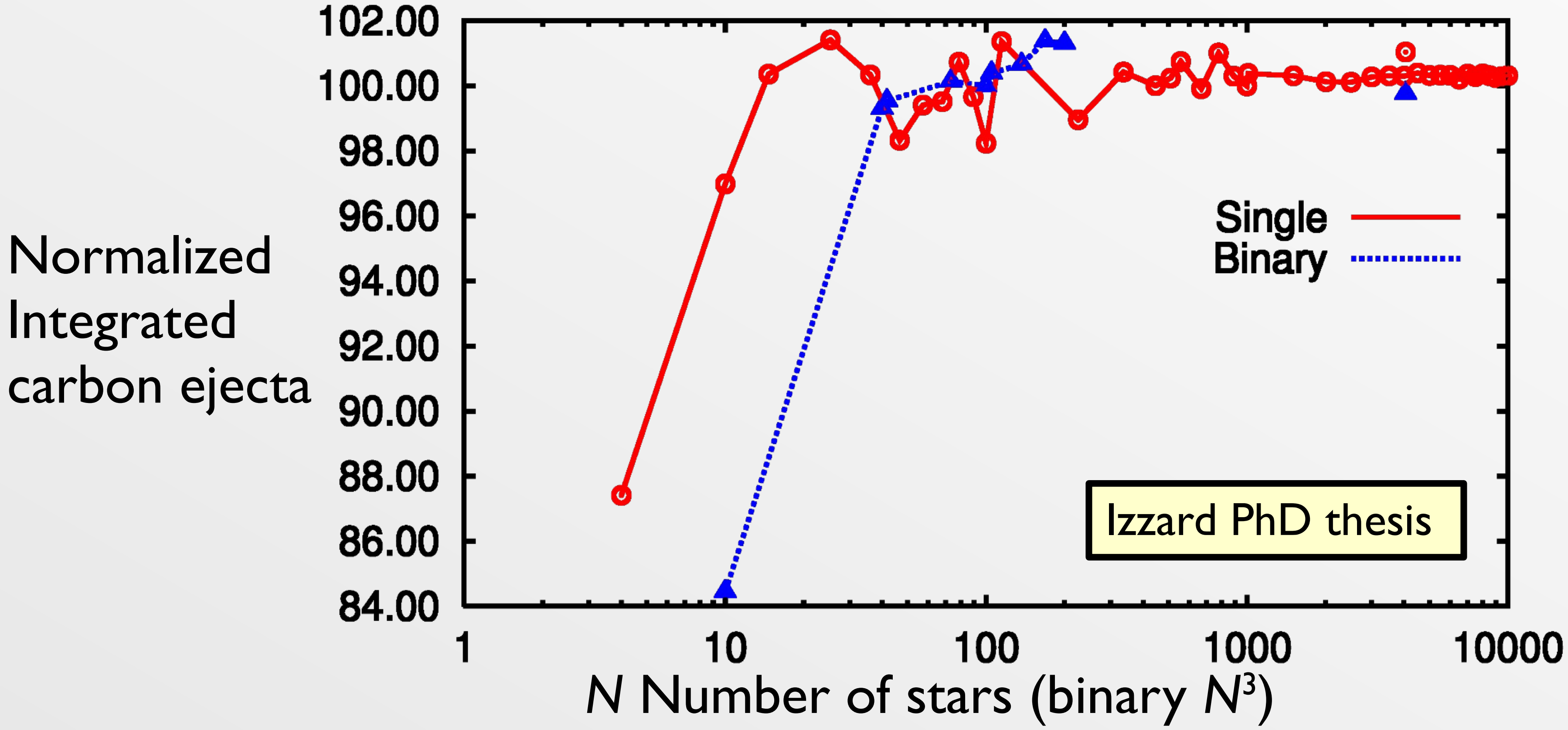
Errors and uncertainties



- Counting errors are **Poisson**

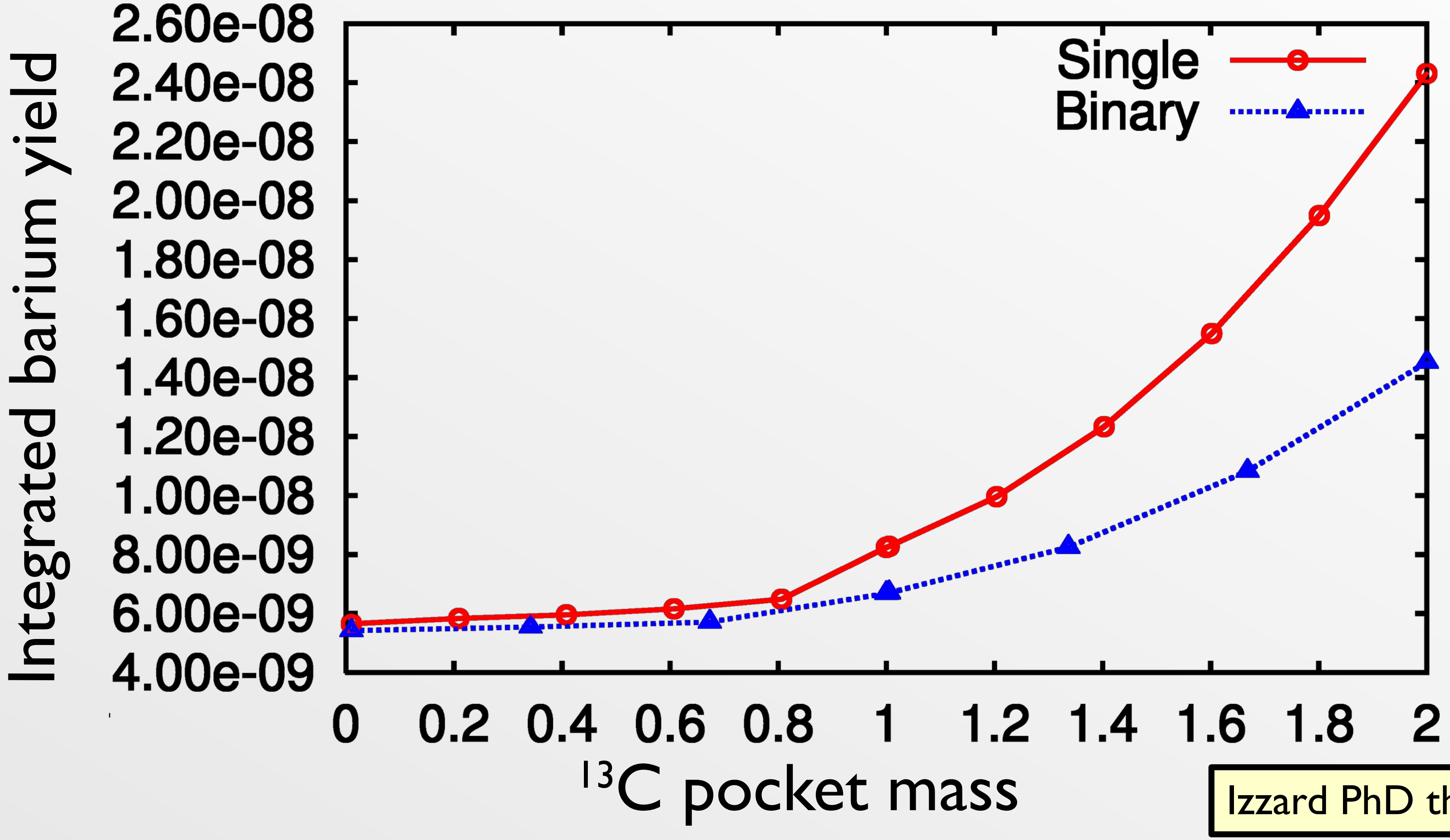
$$\sim \frac{1}{\sqrt{N}}$$

- Solution : **increase *N***



Systematic errors

- Uncertainty in input distributions, model input, etc.



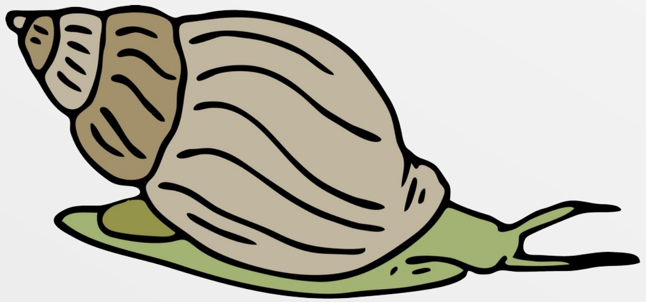
Izzard PhD thesis

Fast and Slow parameters

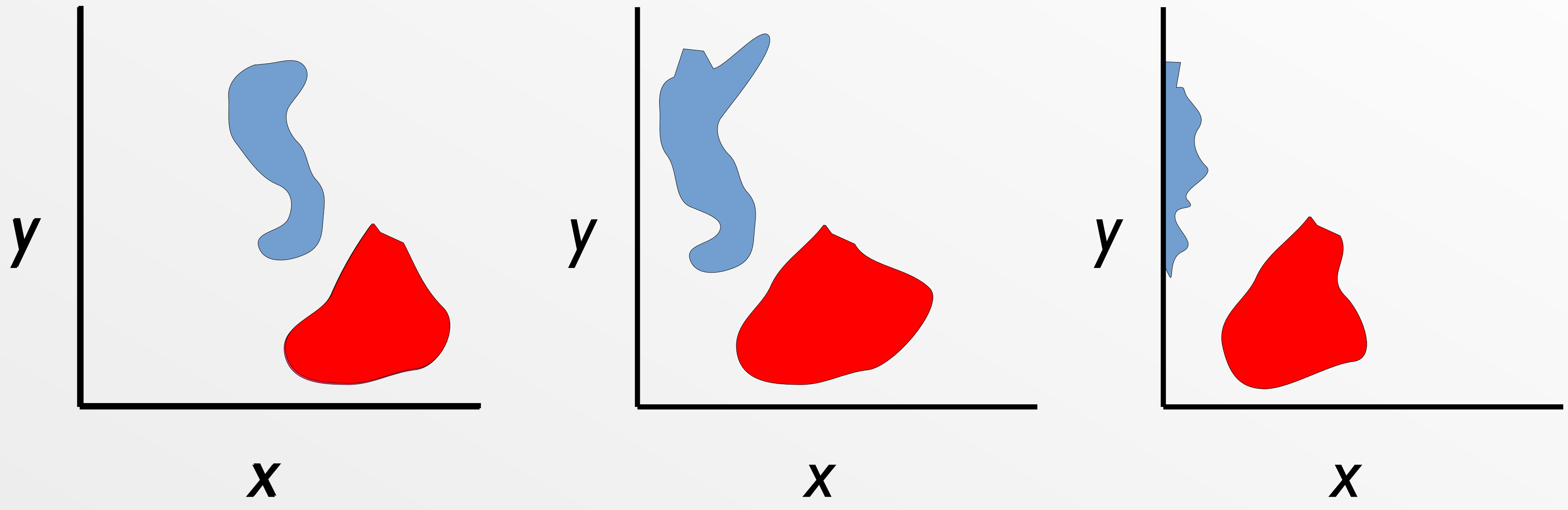
$$n_i = \psi_i \times \Delta t_i$$

Fast parameter:
Given the stellar evolution **only this function** needs to be recalculated.

Slow parameter:
For change in parameter **all the stellar evolution** needs recalculation.



Parameter space blobs



Z=0.001 *Z*=0.005 *Z*=0.01

Many blobs are **red**: they change little with the parameter

Blue example: CH stars – lots at *Z*=0.001; none at *Z*=0.02

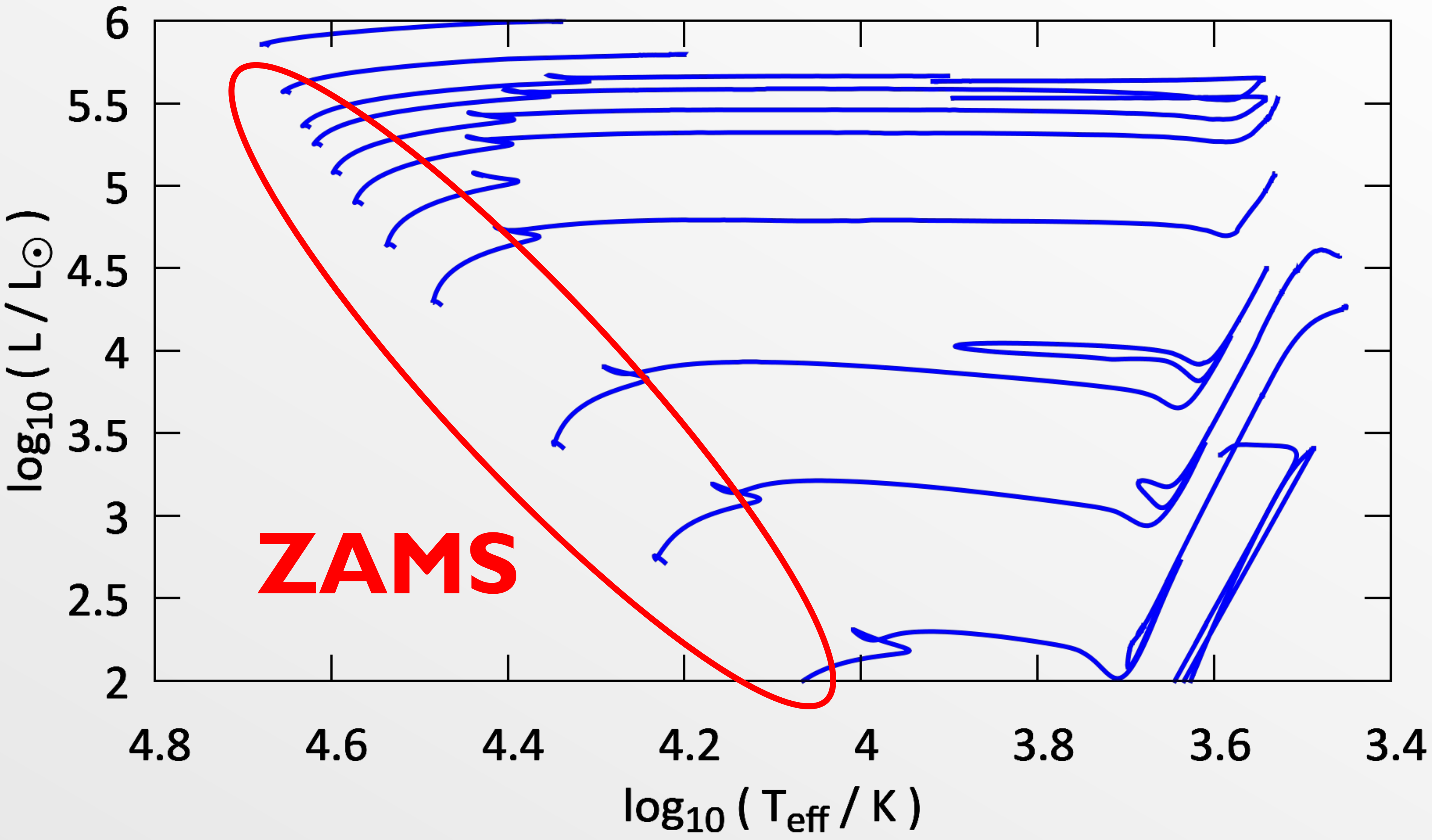
Stellar Evolution Models

- $2N^3$ hours = 2x100x100x100 hours

  ...**at least!**

- Stellar evolution codes are too **slow** and **unreliable** for the task.
- **Need another solution:**
  - **Synthetic stellar evolution** models
  - **Fast**, perhaps **approximate**, codes
  - Need **full stellar codes** for their input!

Example: Zero-age main sequence



TWIN models made with ***Window to the Stars*** in ~ few minutes

http://personal.ph.surrey.ac.uk/~ri0005/window.html

Example: Zero-age main sequence

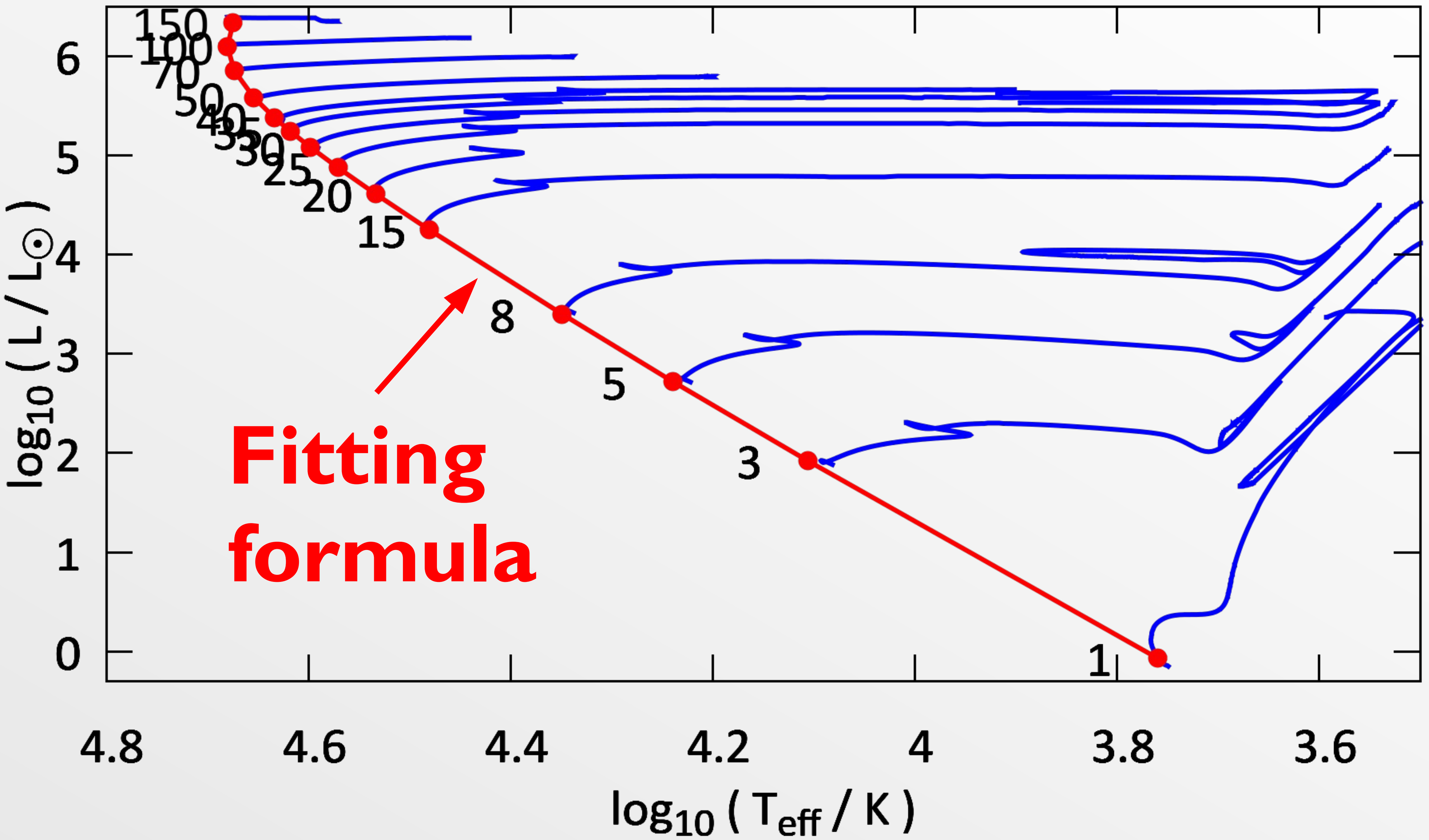
- Eggleton, Fitchett, Tout 1989, Hurley et al 2000, 2002

$$L_0 = \begin{cases} \dfrac{1.107M^3+240.7M^9}{1+281.9M^4} & M \leq 1.093 \\ \dfrac{13990M^5}{M^4+2151M^2+3908M+9536} & M \geq 1.093 \end{cases}$$

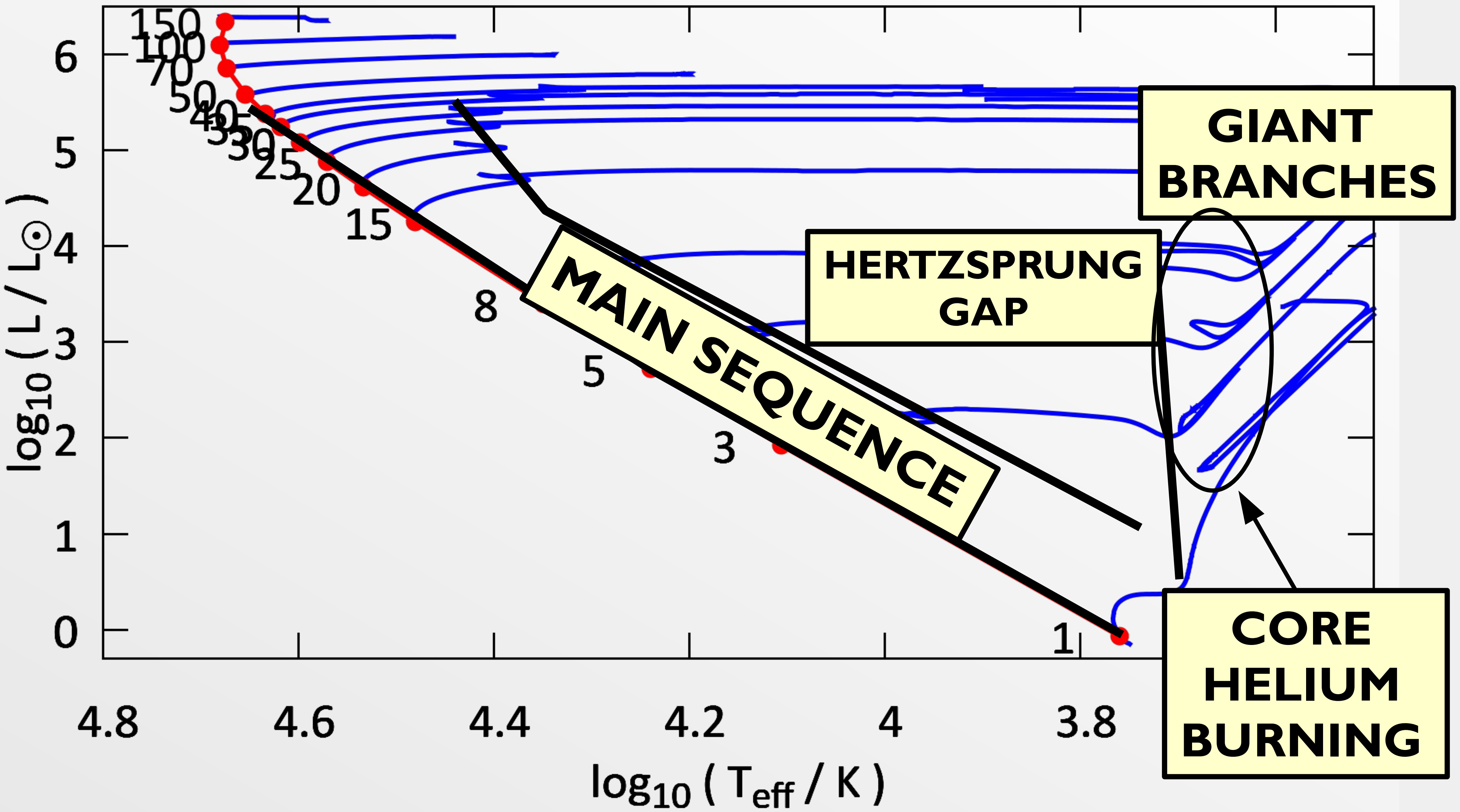
$$R_0 = \begin{cases} \dfrac{0.1148M^{1.25}+0.8604M^{3.25}}{0.04651+M^2} & M \leq 1.334 \\ \dfrac{1.968M^{2.887}-0.7388M^{1.679}}{1.821M^{2.337}-1} & M \geq 1.334 \end{cases}$$

"Simple" formulae : **fast** to calculate.
About 10^6 times faster than a detailed code.

Example: Zero-age main sequence

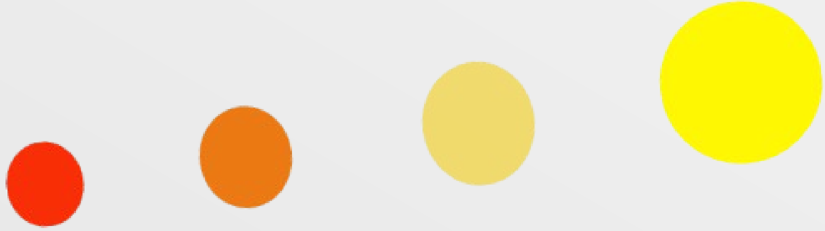
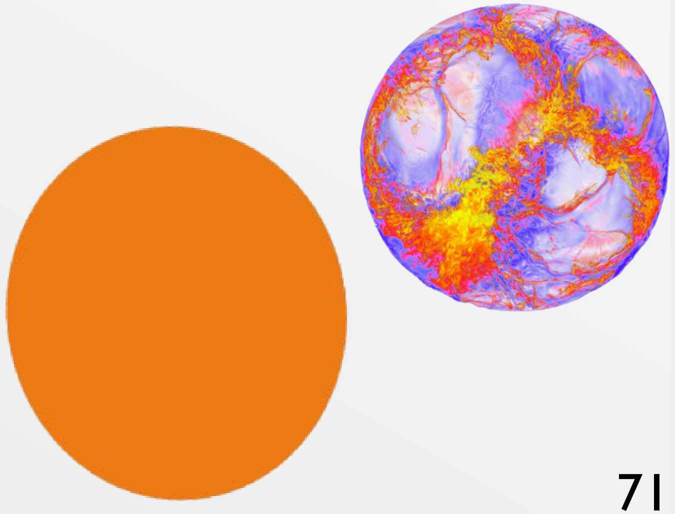


Extend over all HRD



Stellar evolution phases

- 0,1 : Main sequence: convective or radiative
- 2 : Hertzsprung gap: fast, but important for binaries
- 3 : (First) Giant branch: shell hydrogen burning
- 4 : Core helium burning
- 5,6 : AGB: early and thermally pulsing
- 7,8,9 : Stripped stars (binary/wind): Helium stars
- 10,11,12: White dwarfs
- 13,14: Neutron stars, black holes

Phased evolution

- Each phase of evolution has an associated lifetime, e.g.

$$t_{MS} = \frac{2550 + 669M^{2.5} + M^{4.5}}{0.0327M^{1.5} + 0.346M^{4.5}}$$

$$0 \leq \tau = t/t_{MS} \leq 1$$

$$\log_{10} L = \log_{10} L_0 + \alpha\tau_{MS} + \beta\tau_{MS}^2$$

$$\log_{10} R = \log_{10} R_0 + \alpha'\tau_{MS} + \beta'\tau_{MS} + \gamma'\tau_{MS}^3$$

Constants are functions of **mass, metallicity**

$$\alpha = \begin{cases} 0.2594 + 0.1348 \log_{10} M & M \leq 1.334 \\ 0.09209 + 0.05934 \log_{10} M & M > 1.334 \end{cases}$$

$$\beta = \begin{cases} 0.144 - 0.833 \log_{10} M & M \leq 1.334 \\ 0.3756 \log_{10} M - 0.1744 (\log_{10} M)^2 & M > 1.334 \end{cases}$$

$$\alpha' = \begin{cases} 0 & M \leq 1.334 \\ 0.1509 + 0.1709 \log_{10} M & M > 1.334 \end{cases}$$

$$\beta' = \begin{cases} 0.2226 \log_{10} M & M \leq 1.334 \\ -0.4805 \log_{10} M & M > 1.334 \end{cases}$$

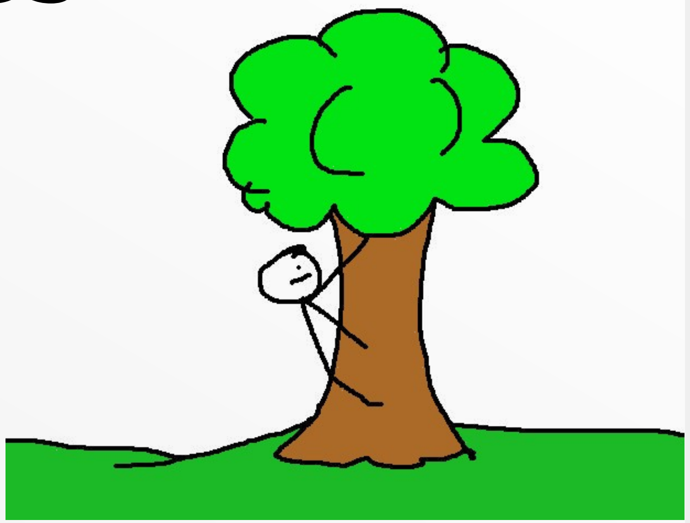
$$\gamma' = \begin{cases} 0.1151 & M \leq 1.334 \\ 0.5083 \log_{10} M & M > 1.334 \end{cases} .$$

This is just for solar metallicity.
But computers don't care if the code is complicated.

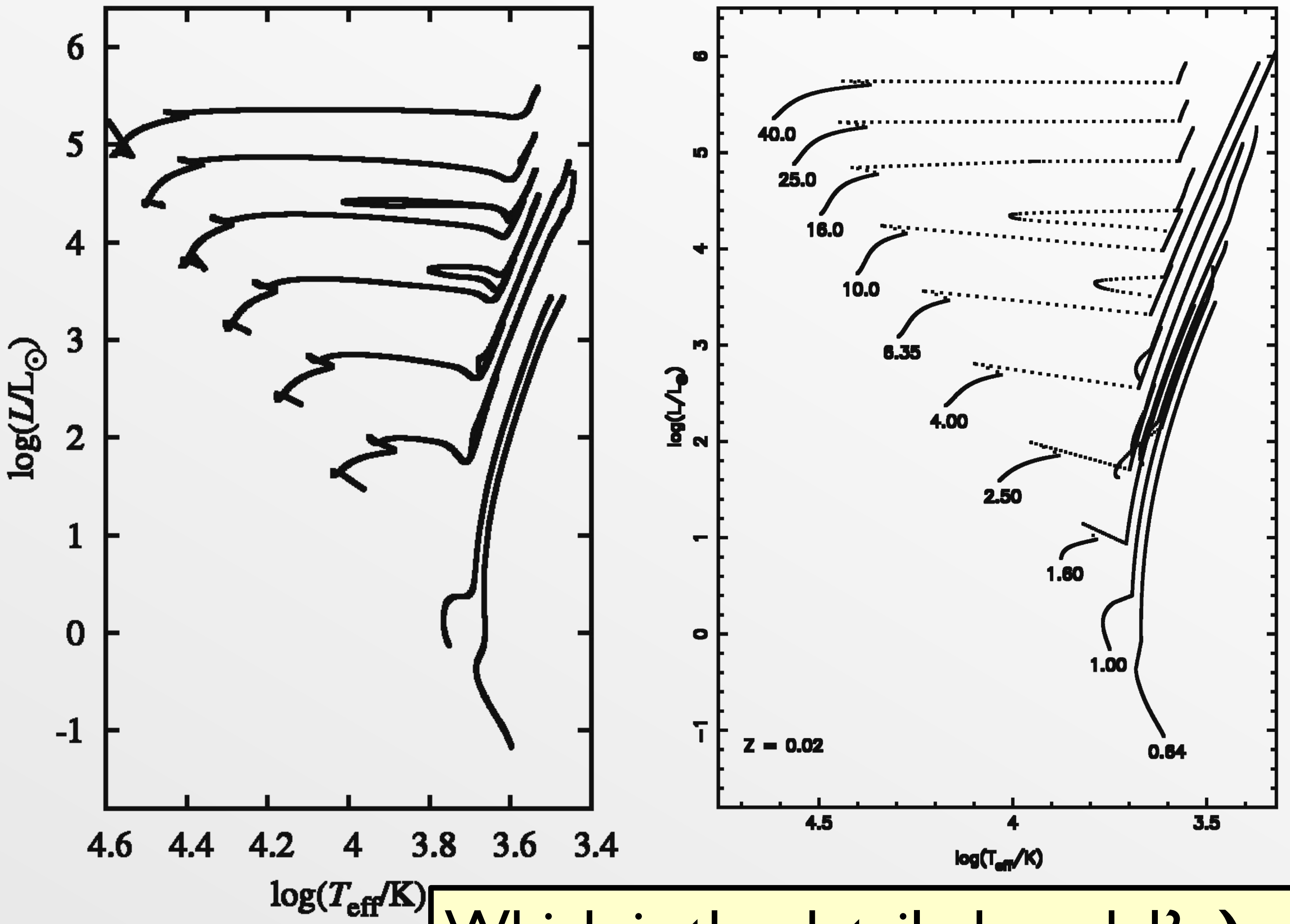
Giant branches

$$L = AM_c^x$$
$$= B\dot{M}_c$$

$$\dot{M}_c = \frac{A}{B} M_c^x$$



So given M_{c0} at the base of the giant branch, the core – hence L and R – evolution can be modelled.



Which is the detailed model? :)

Interpolation libraries

- Latest developments: more flexible
- Tricky to implement *and* be fast
- *binary_c* has the *MINT* library (in development)
- based on MESA grids, similar developed for SSE/BSE
- "Poseidon" code aims to do everything with newly interpolated *binary* grids. Not sure why though… there's little speed to gain by doing this.

- Observations
- Comparison: Observations vs Models
- **Much more on binary stars!**
  - The binary parameter space is huge
  - Analytic/hybrid codes ideal to explore it
- Case studies with *binary_c*
  - What we are doing now
  - The future