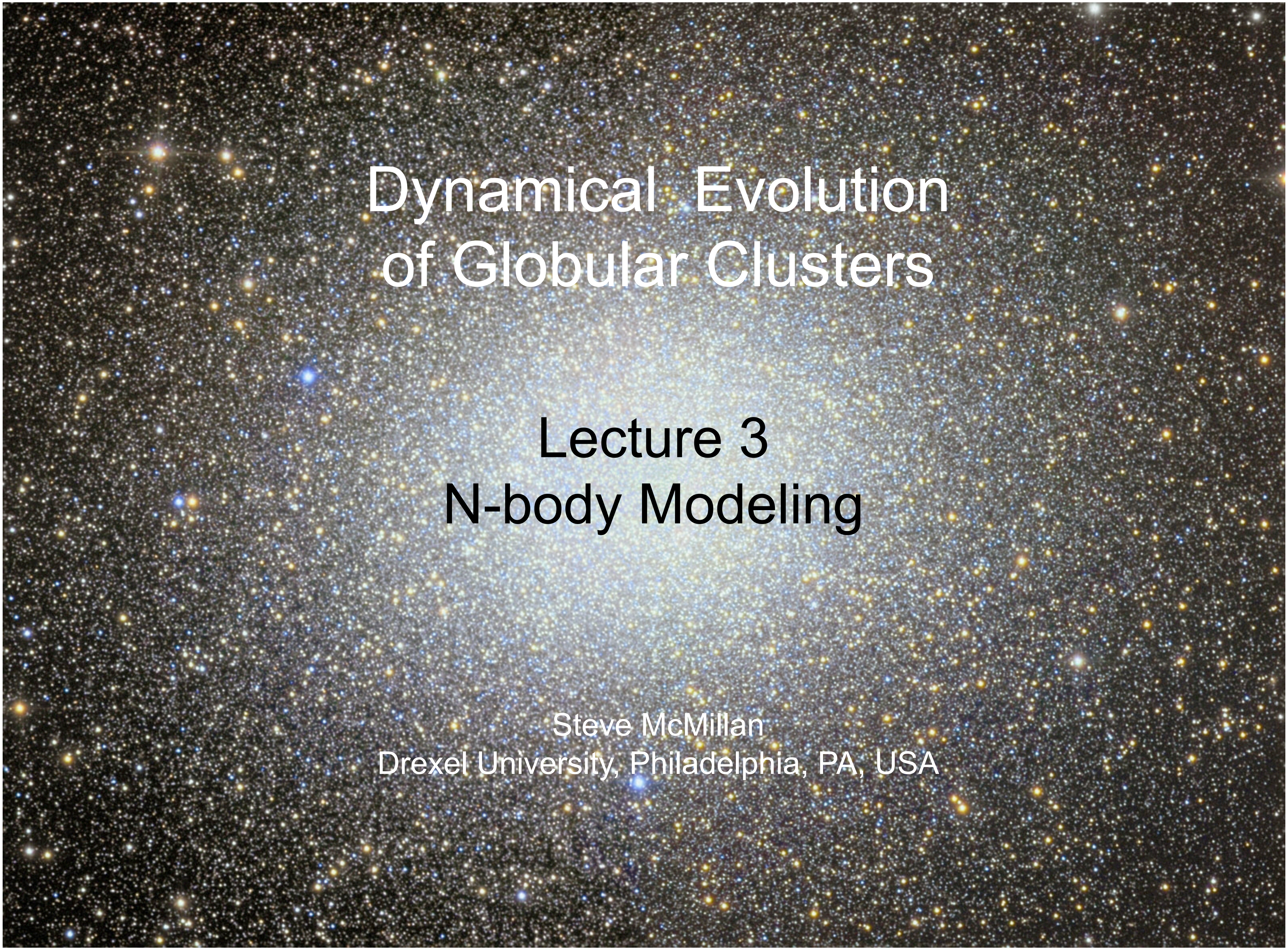


Dynamical Evolution of Globular Clusters

Lecture 3
N-body Modeling

Steve McMillan
Drexel University, Philadelphia, PA, USA

Outline

- fluid and particle methods
- length and time scales
- integration schemes
- evaluation of interparticle forces
- hardware acceleration
- close encounters
- the kichen sink
- the AMUSE project

Gas-Sphere Methods

- stellar dynamical analogs of the familiar equations of stellar structure (mass, hydrostatic equilibrium, energy transport):

$$\frac{\partial M}{\partial r} = 4\pi r^2 \rho$$

$$\frac{\partial}{\partial r}\left(\rho v^2\right) = -\frac{GM\rho}{r^2}$$

$$\frac{\partial}{\partial r}\left(\frac{1}{2}\langle v^2\rangle\right) = -\frac{L}{4\pi\kappa r^2}$$

Fokker–Planck Methods

- diffusion equation in phase space (diffusion coefficients are orbit-averaged relaxation rates):

$$
\begin{aligned}
\frac{\partial N}{\partial t} = & -\frac{\partial}{\partial E}\left[N D_E\right] - \frac{\partial}{\partial J}\left[N D_J\right] \\
& + \frac{\partial^2}{\partial E^2}\left[N D_{EE}\right] + \frac{\partial^2}{\partial J^2}\left[N D_{JJ}\right] \\
& + \frac{\partial^2}{\partial E \partial J}\left[N D_{EJ}\right]
\end{aligned}
$$

N-Body Methods

- direct integration of the equations of motion

$$\mathbf{a}_i \equiv \ddot{\mathbf{x}}_i = \sum_{j \neq i}^{N} G m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3}, \quad i = 1, \ldots, N$$

- incorporation of multiphysics
  - dynamics
  - stellar and binary evolution
  - stellar encounters and collisions
  - external fields
  - gas dynamics
  - radiative transfer

Dynamical Modeling Issues

- large dynamic range
- long-term integration
- long-range forces
- large numbers of particles
- close encounters

Dynamic range

• crossing time scale	1–10 Myr
• relaxation time scale	0.1–10 Gyr = 3×10^{17} s
• core collapse time scale	1–100 Gyr
• evaporation time scale	10–1000 Gyr
• 90° scattering (1 kT binary)	
– length scale:	1–10 AU
– time scale:	1–10 yr
• (MS) stellar collision	10^3–10^4 s
• neutron star binary	< 1 s

Integration Schemes

- predictor–corrector schemes generally preferred!
- second order scheme:

$$
\begin{aligned}
x^p &= x^{(n)} + v^{(n)}\delta t + \tfrac{1}{2}a^{(n)}\delta t^2 \\
v^p &= v^{(n)} + a^{(n)}\delta t \\
a^p &= \mathrm{acc}(x^p) \\
x^{(n+1)} &= x^p \\
v^{(n+1)} &= v^{(n)} + \tfrac{1}{2}(a^p + a^{(n)})\delta t
\end{aligned}
$$

- too inaccurate for use in collisional systems
- widely used in galactic dynamics
- time reversible

Higher Derivatives

- define $$\mathbf{a}_i = \sum_{j\neq i}^{N} \mathbf{a}_{ij}\,, \quad \mathbf{j}_i \equiv \frac{d\mathbf{a}_i}{dt} = \sum_{j\neq i}^{N} \mathbf{j}_{ij}\,, \quad \mathbf{s}_i = \frac{d\mathbf{j}_i}{dt}\,, \quad \text{etc.}$$

where
$$\begin{aligned}
\mathbf{a}_{ij} &= \frac{Gm_j\mathbf{r}_{ij}}{r_{ij}^3} \\
\mathbf{j}_{ij} &= \frac{Gm_j\mathbf{v}_{ij}}{r_{ij}^3} - 3\alpha_{ij}\mathbf{a}_{ij} \\
\mathbf{s}_{ij} &= \frac{Gm_j\mathbf{a}_{ij}}{r_{ij}^3} - 6\alpha_{ij}\mathbf{j}_{ij} - 3\beta_{ij}\mathbf{a}_{ij} \\
\mathbf{c}_{ij} &= \frac{Gm_j\mathbf{j}_{ij}}{r_{ij}^3} - 9\alpha_{ij}\mathbf{s}_{ij} - 9\beta_{ij}\mathbf{j}_{ij} - 3\gamma_{ij}\mathbf{a}_{ij}
\end{aligned}$$

and
$$\begin{aligned}
\alpha_{ij} &= \frac{\mathbf{r}_{ij}\cdot\mathbf{v}_{ij}}{r_{ij}^2}\,, \quad \beta_{ij} = \frac{v_{ij}^2 + \mathbf{r}_{ij}\cdot\mathbf{a}_{ij}}{r_{ij}^2} + \alpha_{ij}^2 \\
\gamma_{ij} &= \frac{3\mathbf{v}_{ij}\cdot\mathbf{a}_{ij} + \mathbf{r}_{ij}\cdot\mathbf{j}_{ij}}{r_{ij}^2} + \alpha_{ij}(3\beta_{ij} - 4\alpha_{ij}^2)
\end{aligned}$$

Fourth-Order Hermite Scheme

$$
\begin{aligned}
x^p &= x^{(n)} + v^{(n)}\delta t + \tfrac{1}{2}a^{(n)}\delta t^2 \ + \tfrac{1}{6}j^{(n)}\delta t^3 \\
v^p &= v^{(n)} + a^{(n)}\delta t + \tfrac{1}{2}j^{(n)}\delta t^2 \\
a^p &= \mathrm{acc}(x^p) \\
j^p &= \mathrm{jerk}(x^p, v^p) \\
v^{(n+1)} &= v^{(n)} + \tfrac{1}{2}(a^{(n)} + a^p)\delta t \ + \tfrac{1}{12}(j^{(n)} - j^p)\delta t^2 \\
x^{(n+1)} &= x^{(n)} + \tfrac{1}{2}(v^{(n)} + v^{(n+1)})\delta t \ + \tfrac{1}{12}(a^{(n)} - a^p)\delta t^2
\end{aligned}
$$

(Makino & Aarseth 1992)

Sixth-Order Hermite Scheme

$$\begin{aligned}
x^p &= x^{(n)} + v^{(n)}\delta t + \tfrac{1}{2}a^{(n)}\delta t^2 + \tfrac{1}{6}j^{(n)}\delta t^3 + \tfrac{1}{24}s^{(n)}\delta t^4 + \tfrac{1}{120}c^{(n)}\delta t^5 \\
v^p &= v^{(n)} + a^{(n)}\delta t + \tfrac{1}{2}j^{(n)}\delta t^2 + \tfrac{1}{6}s^{(n)}\delta t^3 + \tfrac{1}{24}c^{(n)}\delta t^4 \\
a^p &= a^{(n)} + s^{(n)}\delta t + \tfrac{1}{2}c^{(n)}\delta t^2 \\
a^p &= \mathrm{acc}(x^p) \\
j^p &= \mathrm{jerk}(x^p, v^p) \\
s^p &= \mathrm{snap}(x^p, v^p, a^p, j^p) \\
v^{(n+1)} &= v^{(n)} + \tfrac{1}{2}(a^{(n)} + a^p)\delta t + \tfrac{1}{10}(j^{(n)} - j^p)\delta t^2 + \tfrac{1}{120}(s^{(n)} + s^p)\delta t^3 \\
x^{(n+1)} &= x^{(n)} + \tfrac{1}{2}(v^{(n)} + v^{(n+1)})\delta t + \tfrac{1}{10}(a^{(n)} - a^p)\delta t^2 + \tfrac{1}{120}(j^{(n)} + j^p)\delta t^3
\end{aligned}$$

(see Nitadori & Makino 2008 for more)

Integration Schemes

- adaptive, individual/block time steps almost always used
- adaptive higher-order schemes very accurate, but generally <u>not</u> time reversible
- address this using <u>time symmetrization</u>

$$\delta t \;=\; \tfrac{1}{2}\left[\tau(t+\delta t)+\tau(t)\right]$$

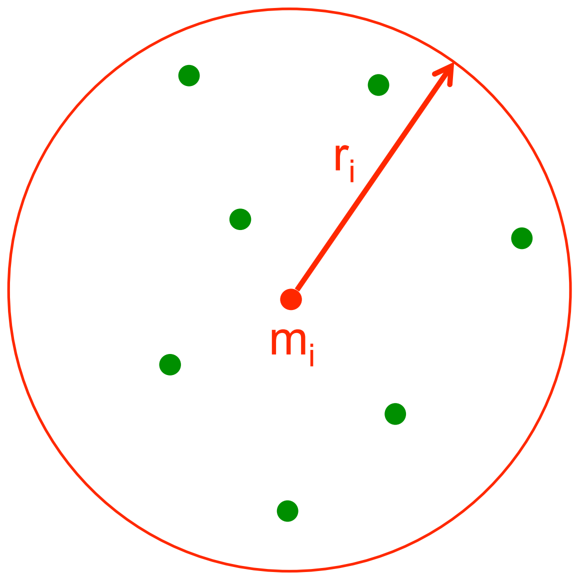
(Hut, Makino, McMillan 1995)

Evaluation of Long-Range Forces

- direct summation (brute force)
- neighbor schemes
- tree codes

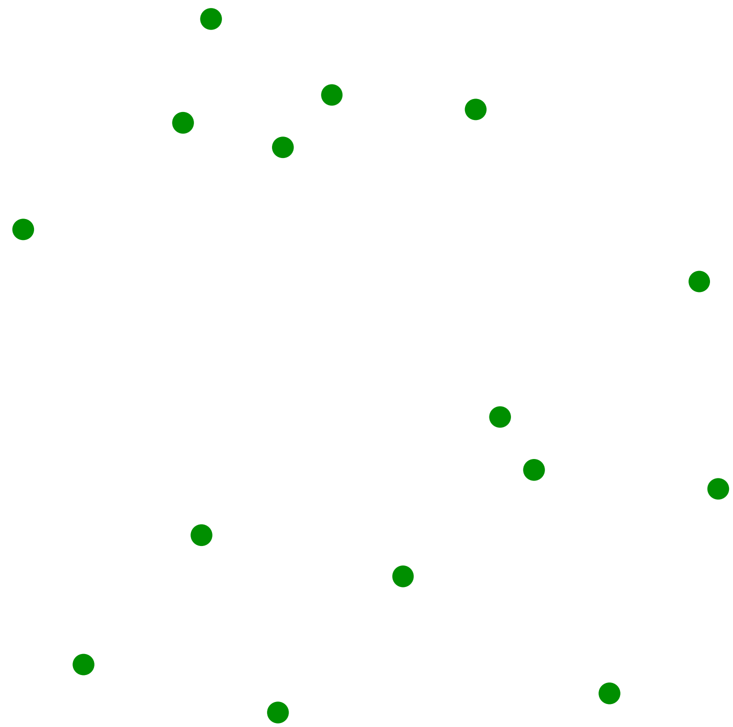
Neighbor Schemes

- Ahmad-Cohen scheme
  - neighbor sphere of radius r_i
  - stars inside the sphere have forces a_{ij} recalculated at every step—<u>irregular</u> forces
  - stars outside have a_{ij} extrapolated at most steps, recalculated on longer time scales—<u>regular</u> forces
  - substantial savings in computation cost over brute-force summation



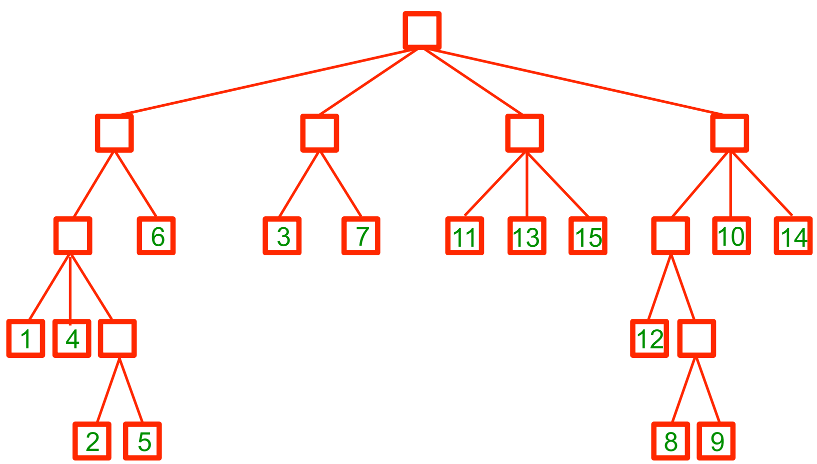
Tree Codes

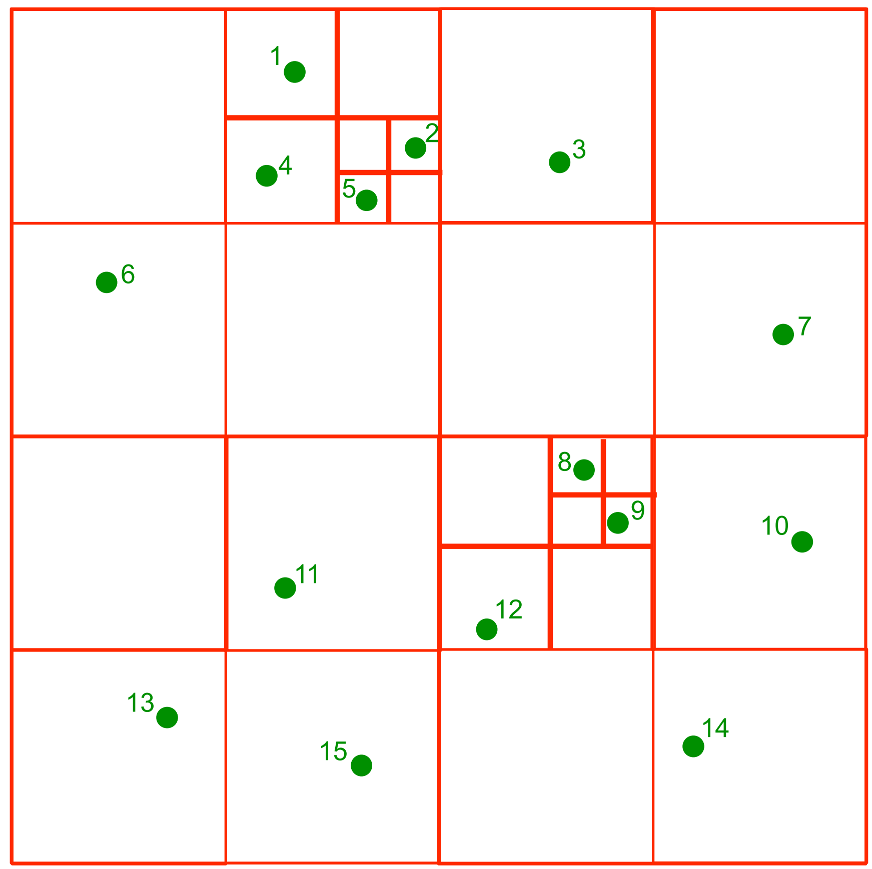
- Barnes-Hut scheme
  - recursively divide space into octants (quadrants) to separate the particles

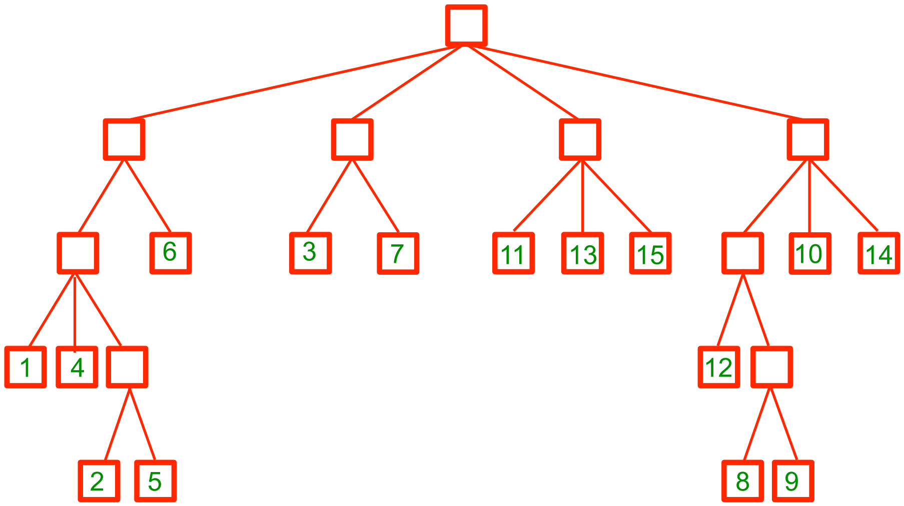


Tree Codes

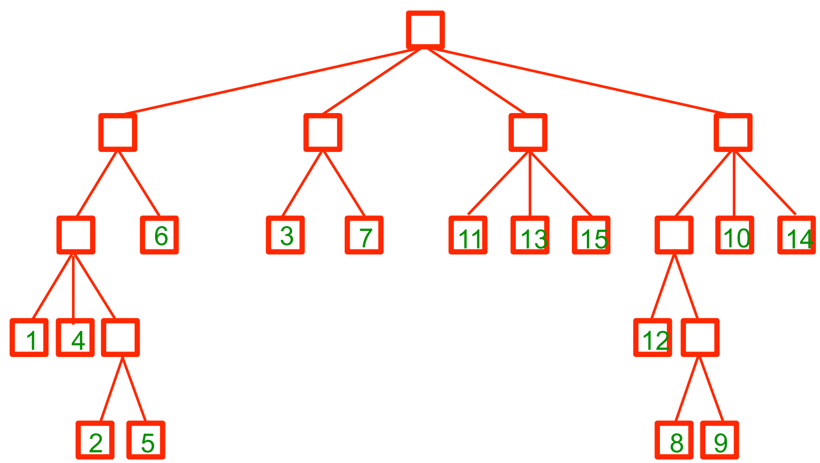
- Barnes-Hut scheme
  - recursively divide space into octants (quadrants) to separate the particles
  - each particle has a unique location in the tree







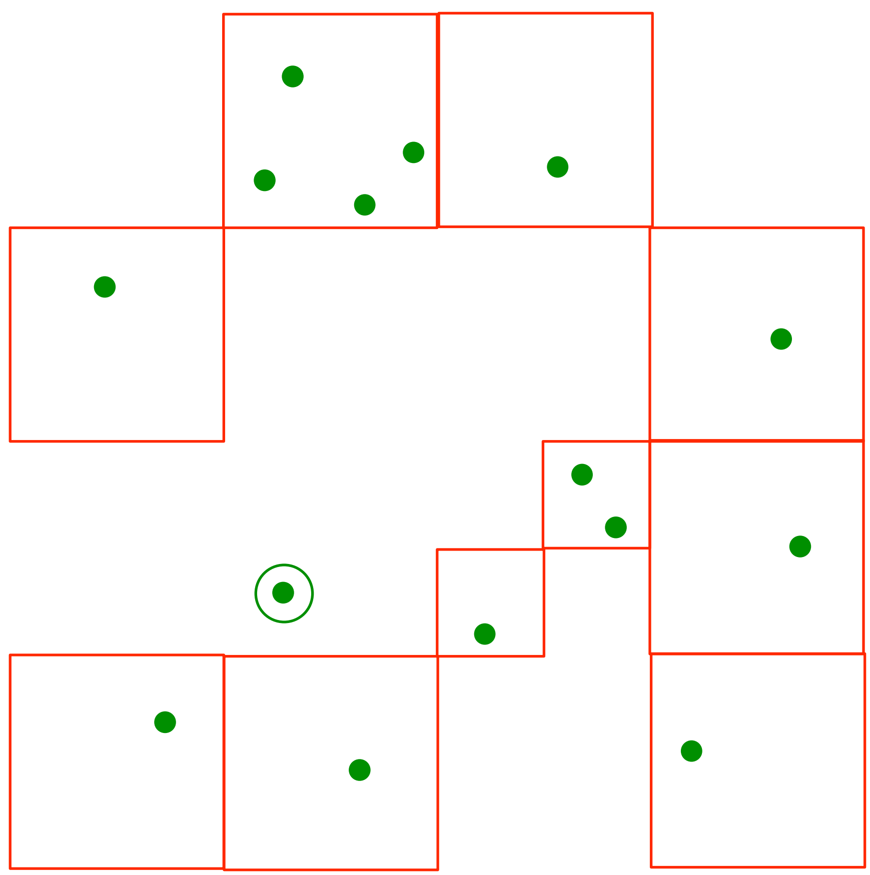
6
3
7
11
13
15
10
14
1
4
12
2
5
8
9



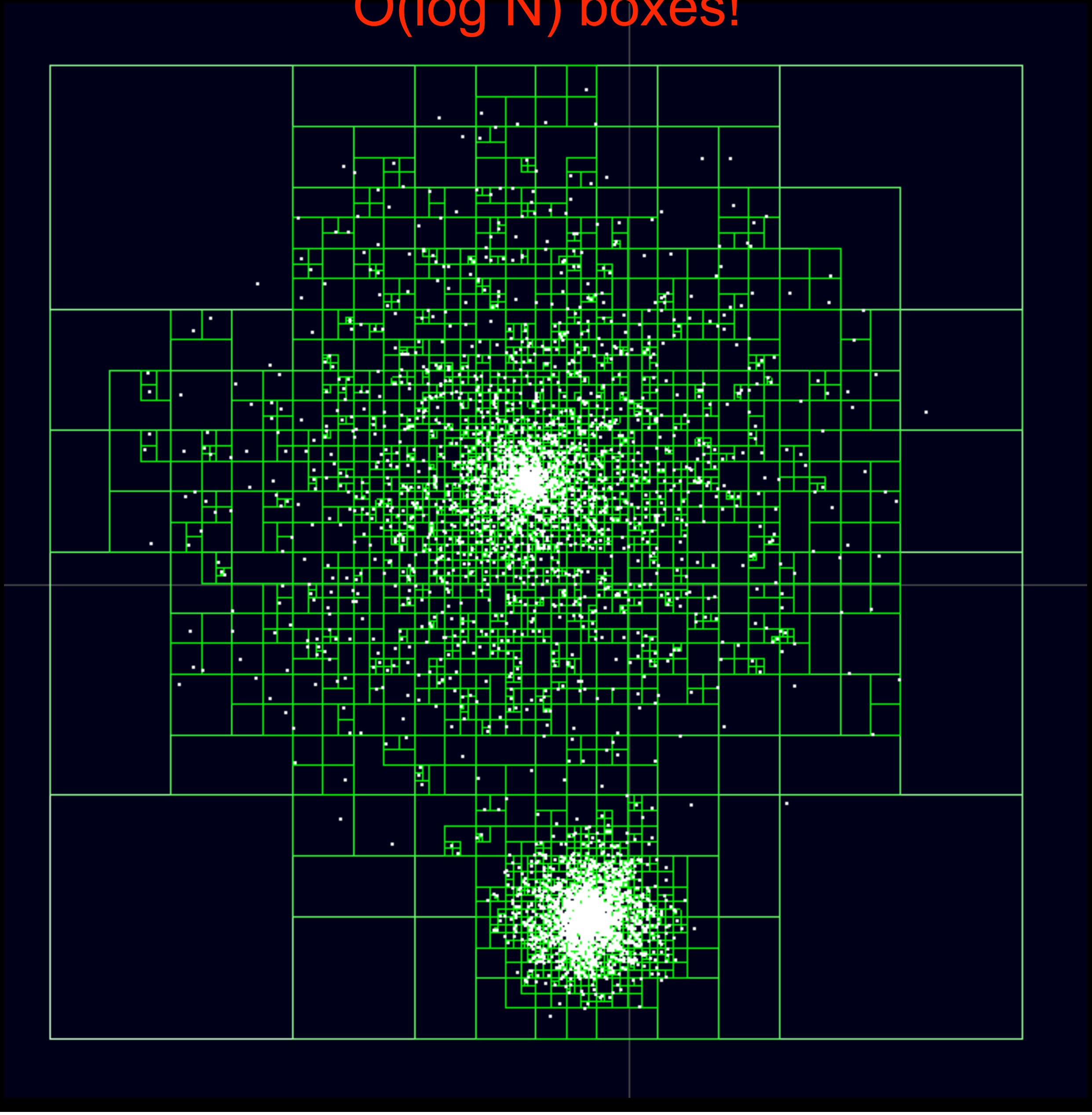
6
3
7
11
13
15
10
14
1
4
12
2
5
8
9

Tree Codes

- Barnes-Hut scheme
  - recursively divide space into octants (quadrants) to separate the particles
  - each particle has a unique location in the tree
  - for each particle on which the force is needed, descend the tree, opening only nearby boxes
  - unopened boxes are treated as a multipole expansion



O(log N) boxes!



Hardware Acceleration

- the GRAPE project

$$\mathbf{a}_i \equiv \ddot{\mathbf{x}}_i = \sum_{j \neq i}^{N} G m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3}, \quad i = 1, \ldots, N$$

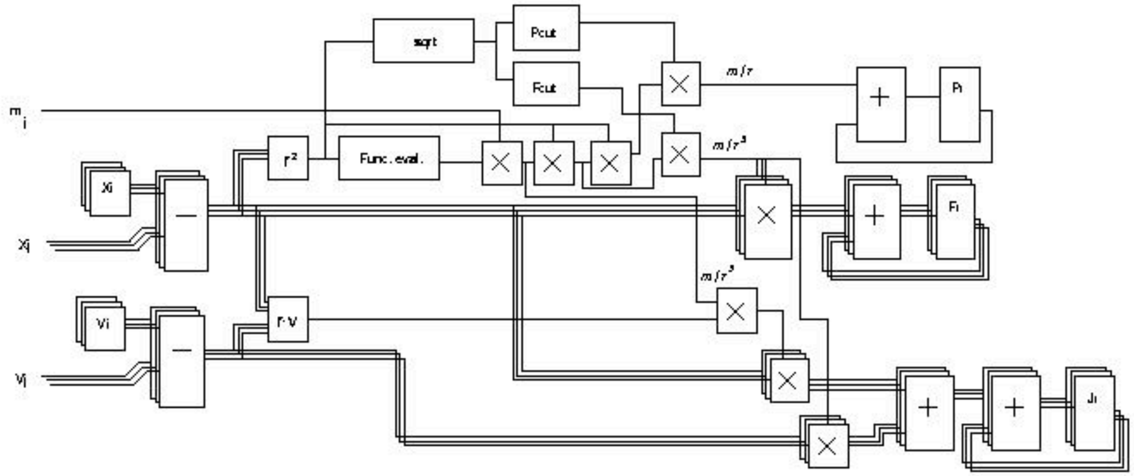
Hardware Acceleration

- the GRAPE project

```
for (int j = 0; j < n; j++)
    if (j != i) {
        double r2 = 0;
        for (int k = 0; k < 3; k++) {
            dx[k] = pos[j][k] - pos[i][k];
            r2 += dx[k]*dx[k];
        }
        double mr3i = mass[j]/(r2/sqrt(r2));
        for (int k = 0; k < 3; k++)
            acc[i][k] += dx[k]*mr3i;
    }
```

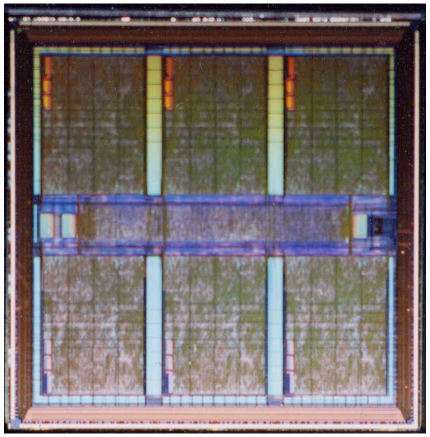
Hardware Acceleration

- the GRAPE project

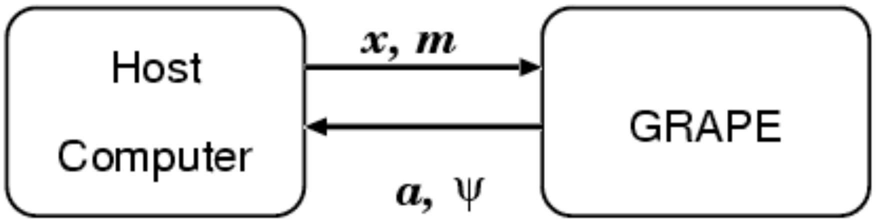


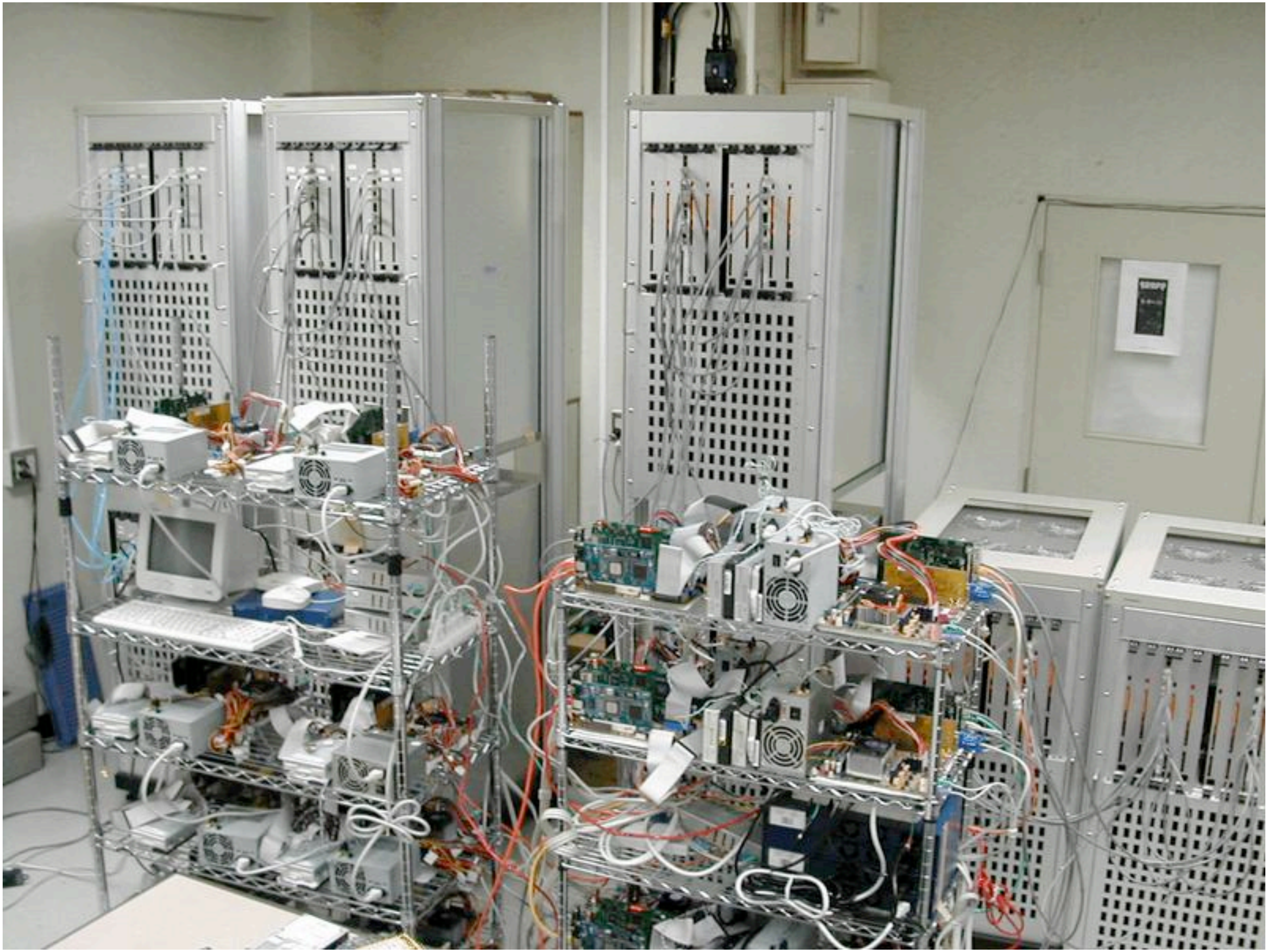
Hardware Acceleration

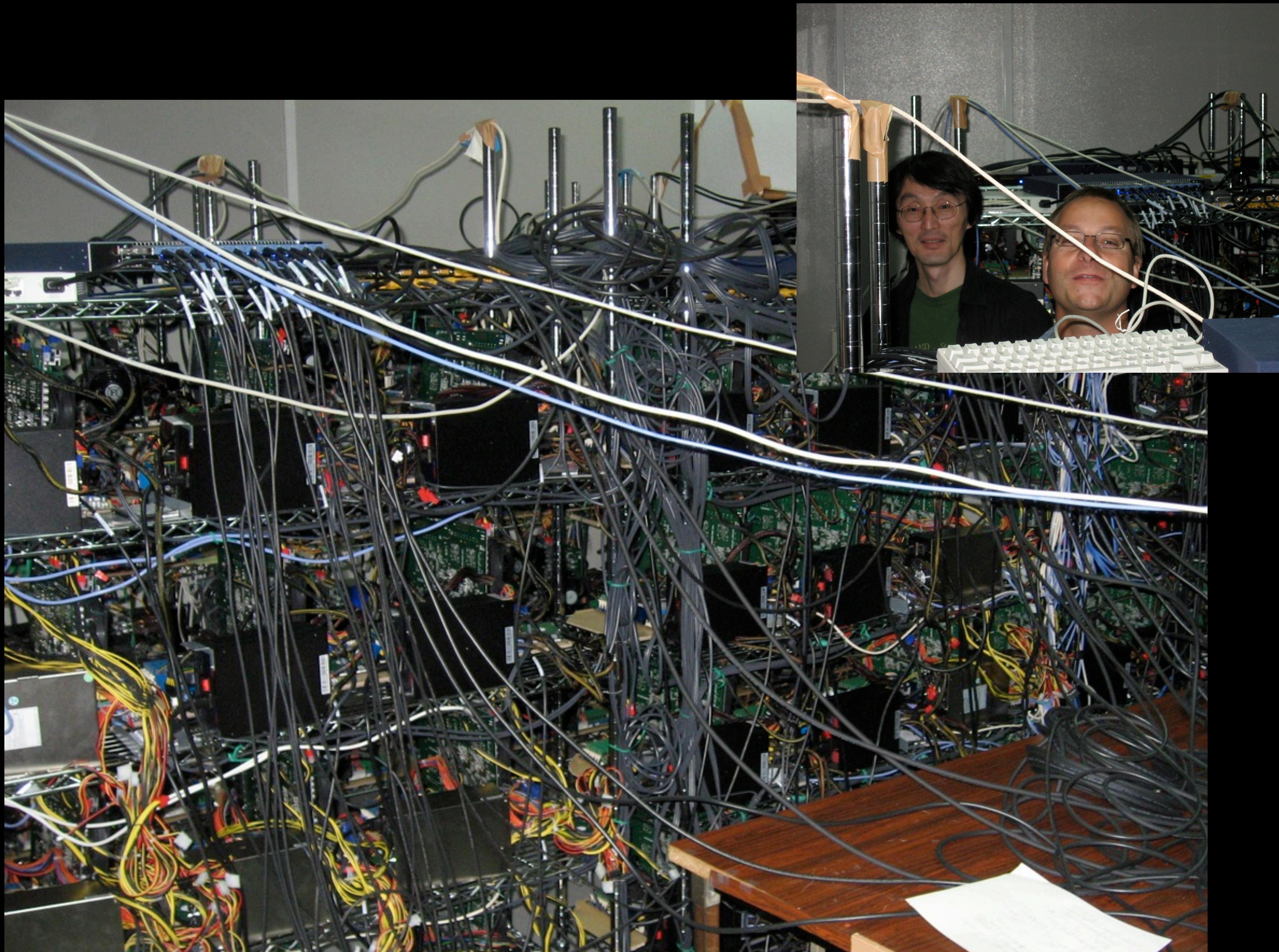
- the GRAPE project



- special-purpose architecture, deeply pipelined, massively parallel (Sugimoto et al.1990 ; Makino et al. 2005)
  - GRAvity PipE
  - gravitational force accelerator

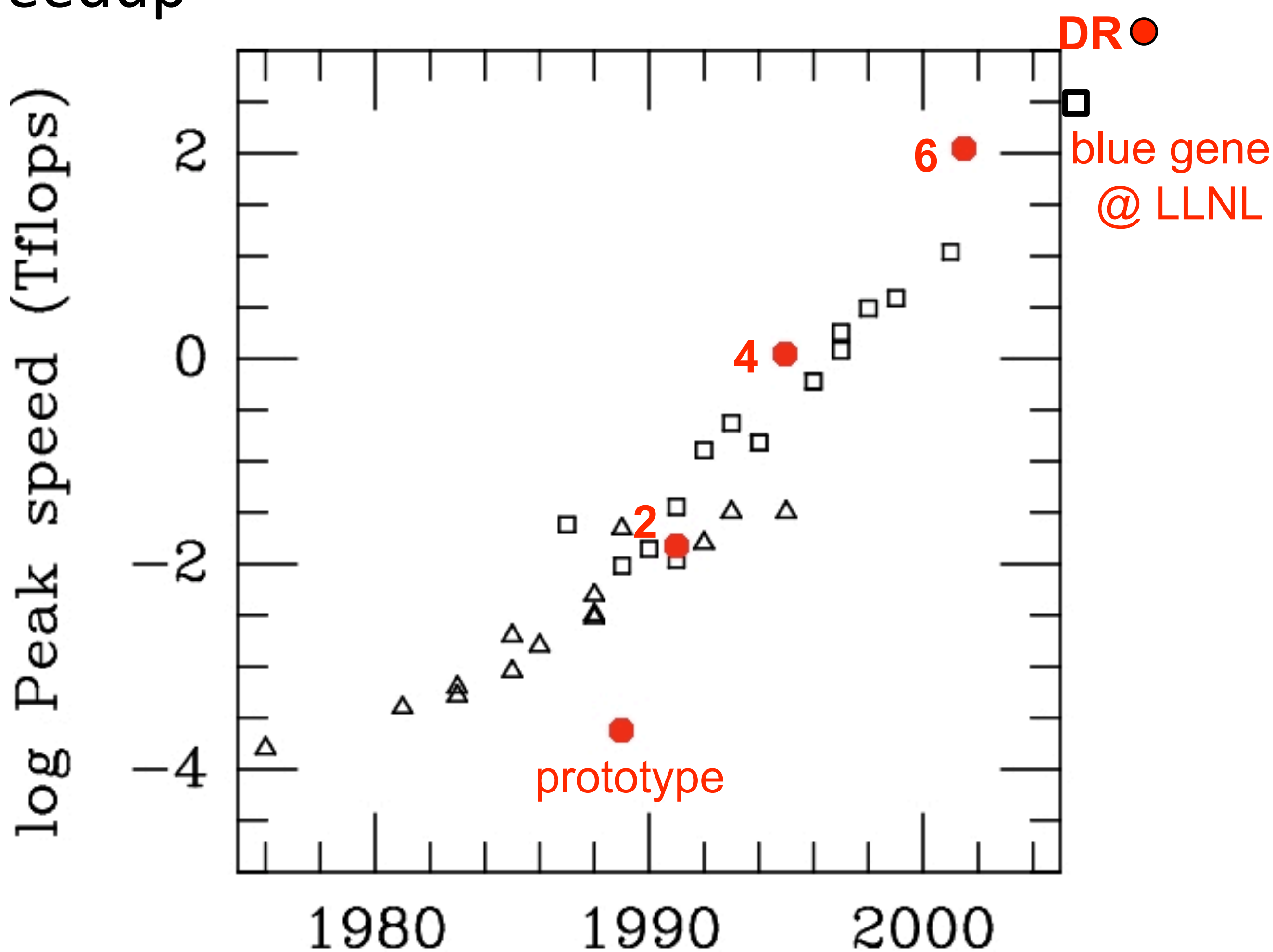






Hardware Acceleration

- GRAPE speedup



Hardware Acceleration

- GRAPE speed ~1 Tflop/s per chip (latest generation DR)
- comparable speeds now achievable with GPUs

- 50–100 Tflop/s computing power now routinely available in commodity GPU clusters

Close Encounters

- gravity is a singular force: expect problems near $r = 0$
- expect ~1 90° encounter in the entire system per dynamical time
- expect an X kT binary to undergo a close encounter (periastron ~ semimajor axis) every X relaxation times

  $\Rightarrow$ average heating rate per binary ~ kT/t_r
- binaries are dynamically important and potentially long lived, and must be managed along with the large-scale motion

Close Encounters

- close encounters not handled well by standard large-scale integrators
  - energy errors accumulate: 10^{-6} per orbit for a 10 kT binary for 10 relaxation times in a 10^6 particle system $\Longrightarrow$ O(1) total error
  - far too many time steps: 100 steps per period for a 10 kT binary $\Longrightarrow$ 300 N^2 steps per relaxation time
- special treatment of close encounters and binary/multiple motion is <u>essential</u>
  - <u>regularization</u> and/or <u>unperturbed</u> motion

Regularization

- avoid errors associated with singular motion by transforming the equations of motion to remove the singularity
- generally involves both a coordinate and a time transformation

Regularization

- e.g. in two dimensions $\mathbf{r} = (x, y)$

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{GM\mathbf{r}}{r^3}$$

- set $z = x + iy$

$$\frac{d^2z}{dt^2} = -\frac{GMz}{|z|^3}$$

- transform

$$z = Z^2, \quad dt = |Z|^2 d\tau$$

$$\Rightarrow \quad \frac{d^2Z}{d\tau^2} = \tfrac{1}{2}hZ$$

where

$$h = \tfrac{1}{2}v^2 - \frac{GM}{r}$$

Regularization

- avoid errors associated with singular motion by transforming the equations of motion to remove the singularity
- generally involves both a coordinate and a time transformation
- "production" versions
  - Kustaanheimo & Stiefel (1965)
  - "chain regularization" (Mikkola & Aarseth 1990, 1993)
  - "algorithmic regularization" (Mikkola & Tanikawa 1999)

Unperturbed Motion

- switch to unperturbed two-body (kepler) orbit for small perturbations

$$\Gamma \equiv \frac{|\Delta \mathbf{a}_{12}| a^2}{G(m_1 + m_2)} \ll 1$$

- couple with robust estimators of stability for triple and higher-order multiple systems (e.g. Mardling 2007)
- only resolve close binaries and multiples when needed—treat as inert particles the rest of the time
- MUST do this whether or not regularization is employed

“Kitchen Sink” Codes

- monolithic design
- very successful for many dynamical problems
- limited physics menu
  - detailed dynamics
  - approximate stellar evolution
  - semi-analytic/heuristic binary evolution
  - cartoon hydrodynamics
- hard to maintain/modify/expand functionality

The State of the Art

- NBODY4/6/6++ + BSE + sticky spheres

  Aarseth, Hurley, Tout, Spurzem,…

- kira + seba + sticky spheres

  McMillan, Portegies Zwart, Hut, Makino

- MC + startrack/BSE + sticky spheres

  Rasio, Fregeau, Gurkan, Belczynski, Kalogera

  – in all cases: SPH/MMAS after the fact (Lombardi, Gaburov)

Software Issues

- star clusters bring together related fields that have traditionally been pursued independently
- multiphysics problems, software integration essential
- don't want to reinvent the wheel
- large legacy code base
- tradition of open source

Software Issues

- star clusters bring together related fields that have traditionally been pursued independently
- multiphysics problems, software integration essential
- don't want to reinvent the wheel
- large legacy code base
- tradition of open source

Design goals

- software framework to connect formerly independent modules
- interoperability: “plug and play”
  - explicitly enable code calibration and comparison
- don’t hard-wire legacy codes!
- don't mandate a programming style or language
- incorporate legacy code by wrapping it
- address inflexibility in current kitchen sink codes

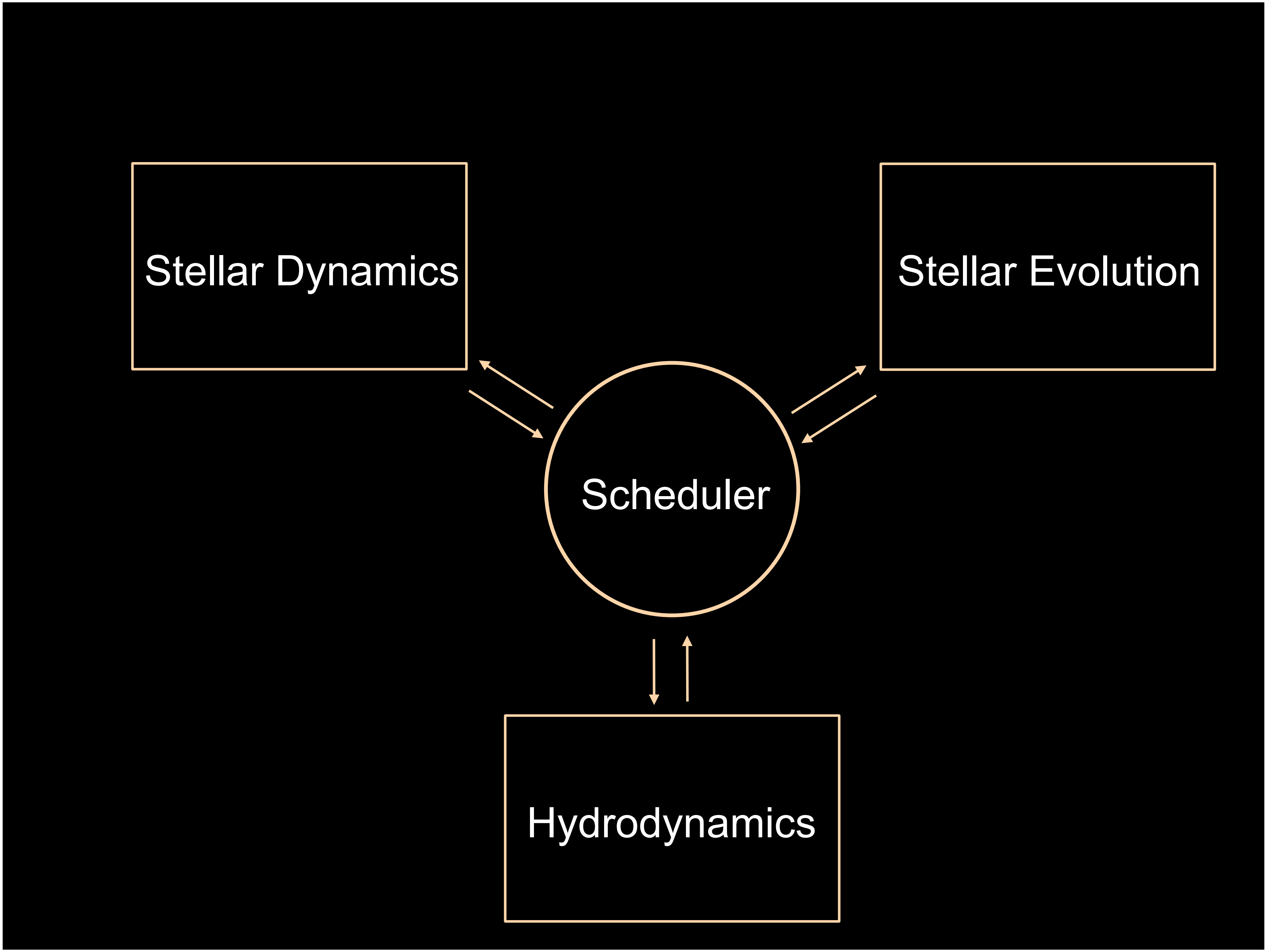
AMUSE

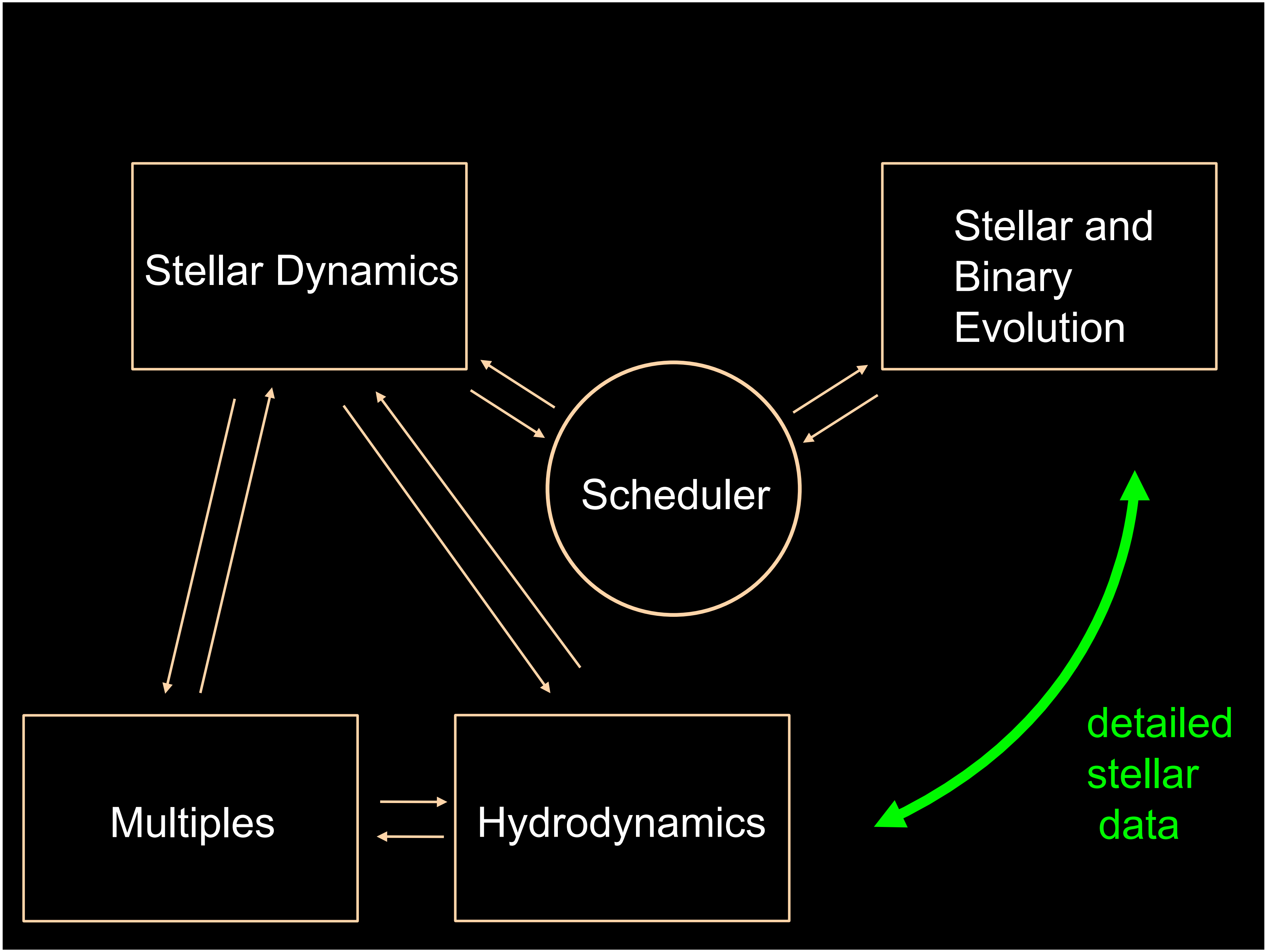
http://amusecode.org

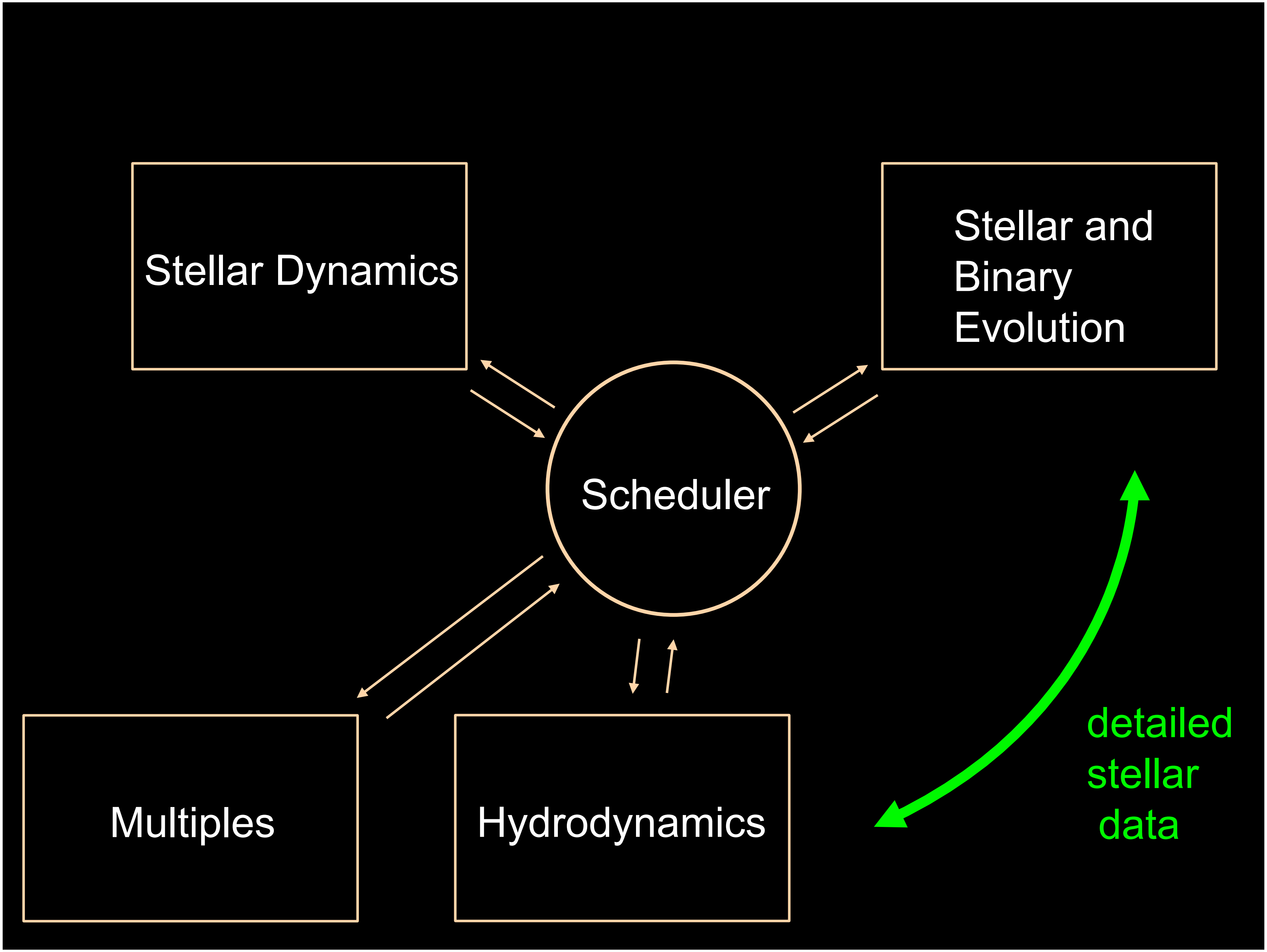
- modules for stars, dynamics, multiples, collisions, gas dynamics, etc.
- implemented as “black boxes” with wrappers
- well defined interfaces
- fully MPI parallel
- use python as a top-level “glue” language

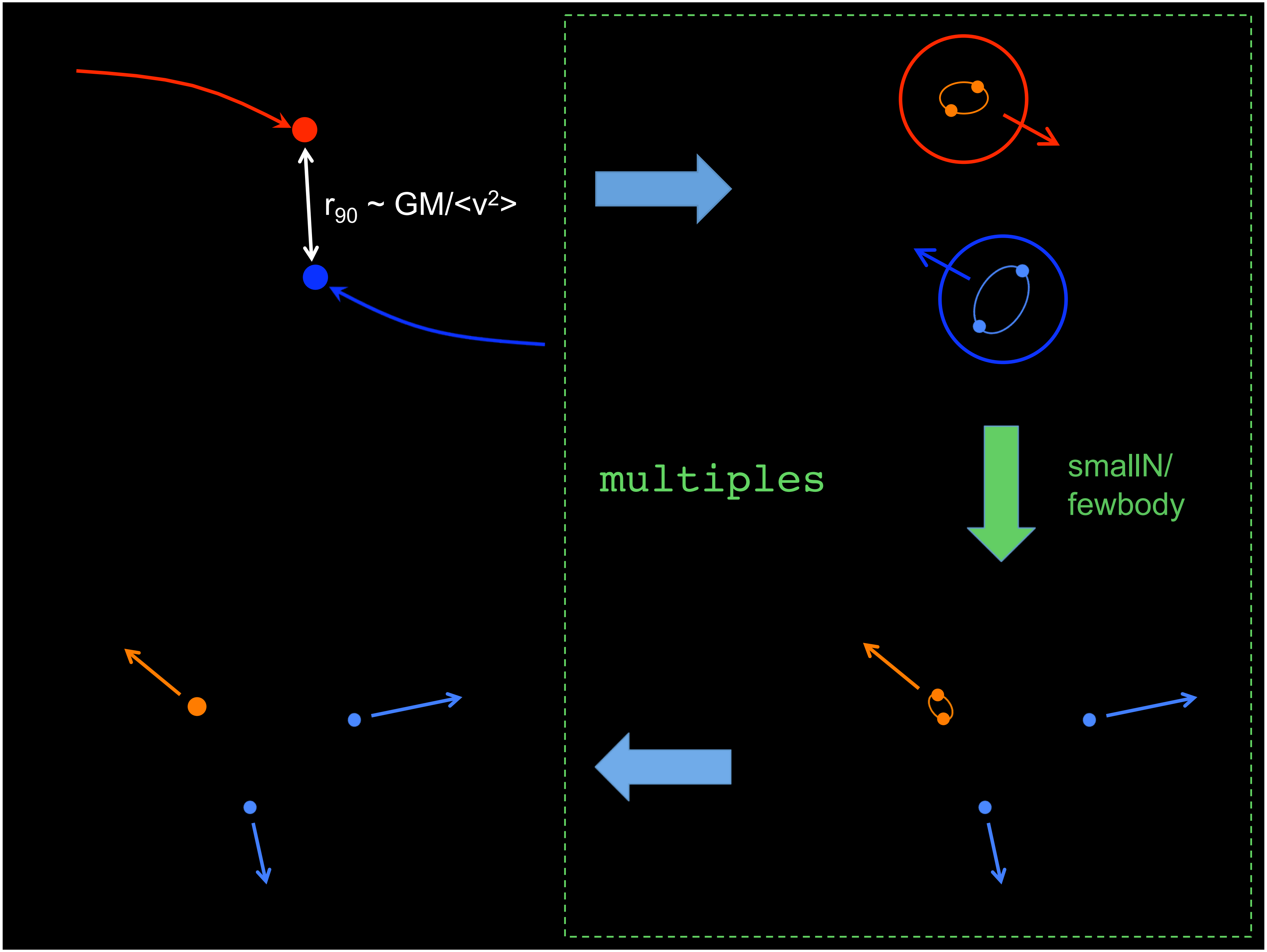
Python as a Glue Language

- flexible
- object oriented
- MPI and hence C, C++, f77, f90, f95,... interfaces
- large user base
- many contributed modules
- numpy acceleration

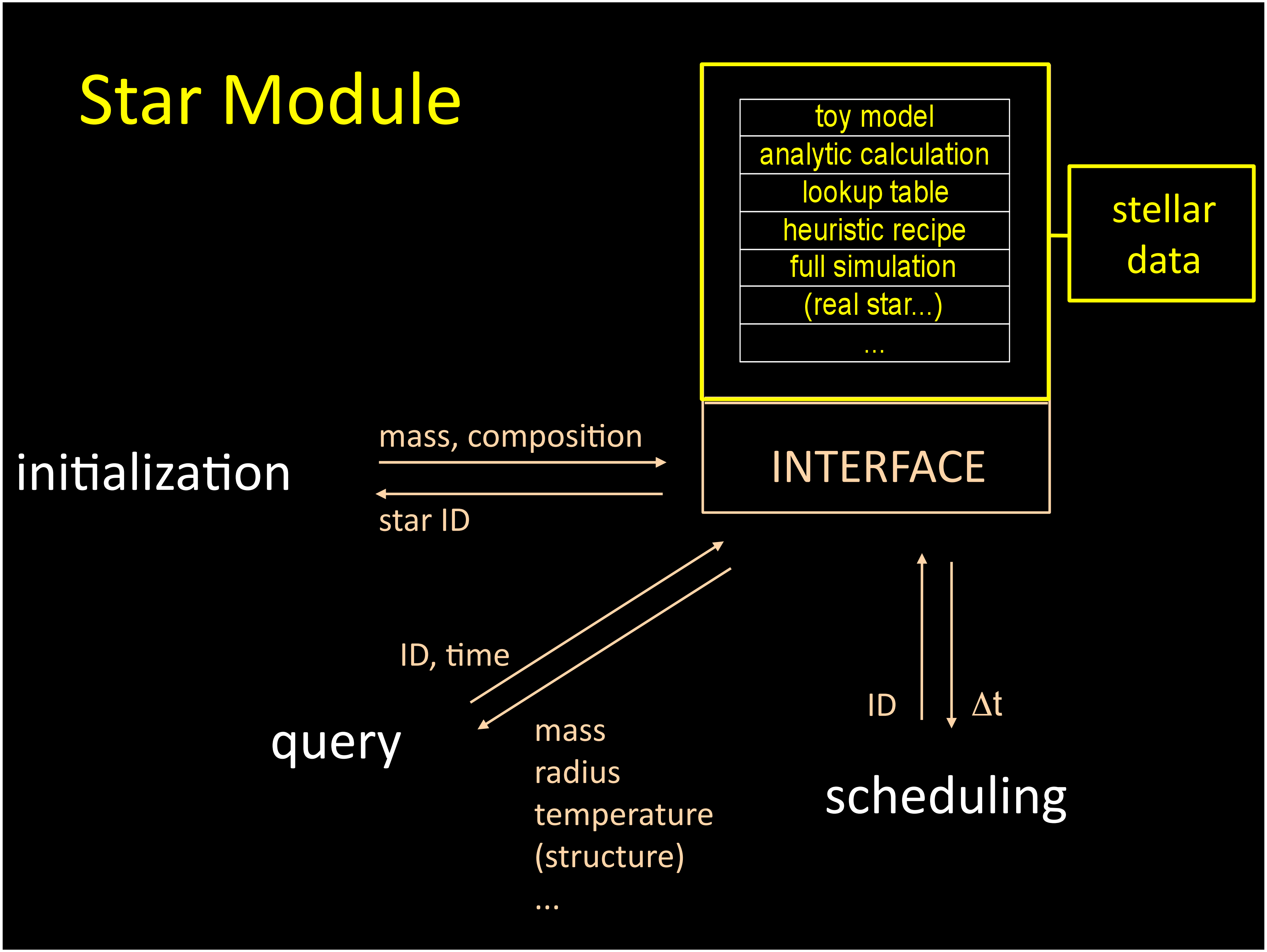

Stellar Dynamics
Stellar Evolution
Scheduler
Hydrodynamics

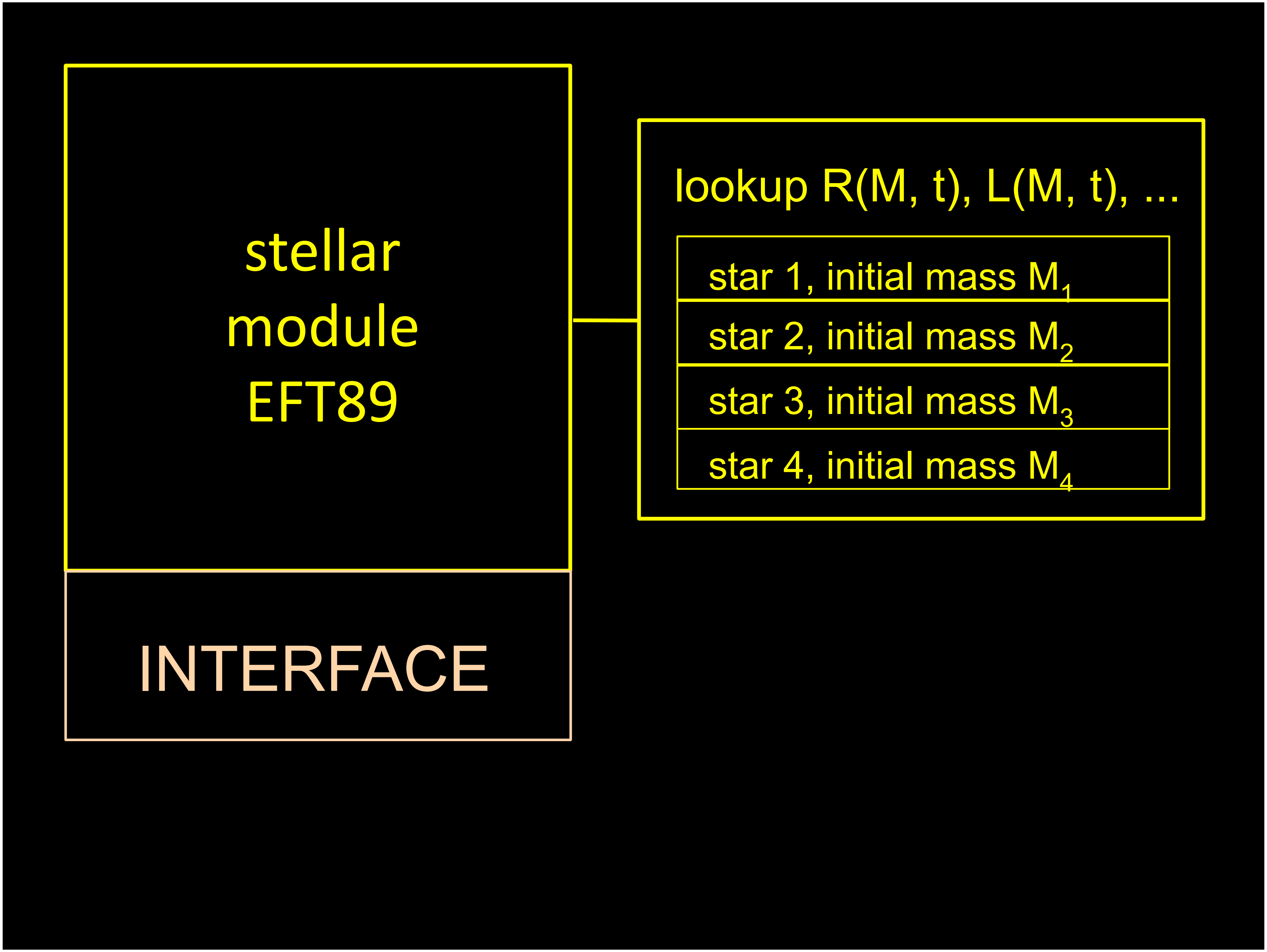

Stellar Dynamics
Stellar and
Binary
Evolution
Scheduler
Multiples
Hydrodynamics
detailed
stellar
data


Stellar Dynamics
Stellar and Binary Evolution
Scheduler
detailed stellar data
Multiples
Hydrodynamics

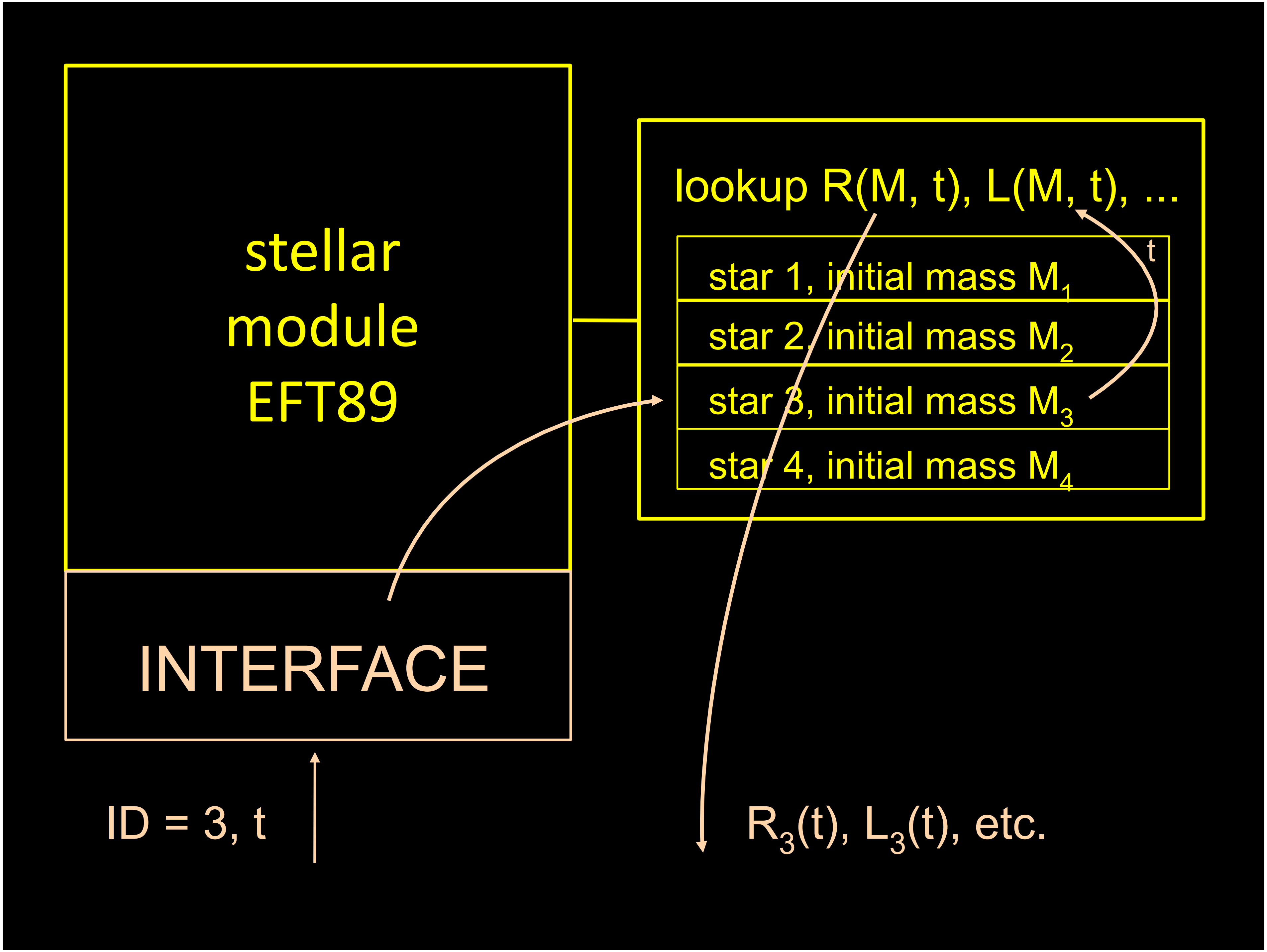


$r_{90} \sim GM/\langle v^2\rangle$
multiples
smallN/
fewbody

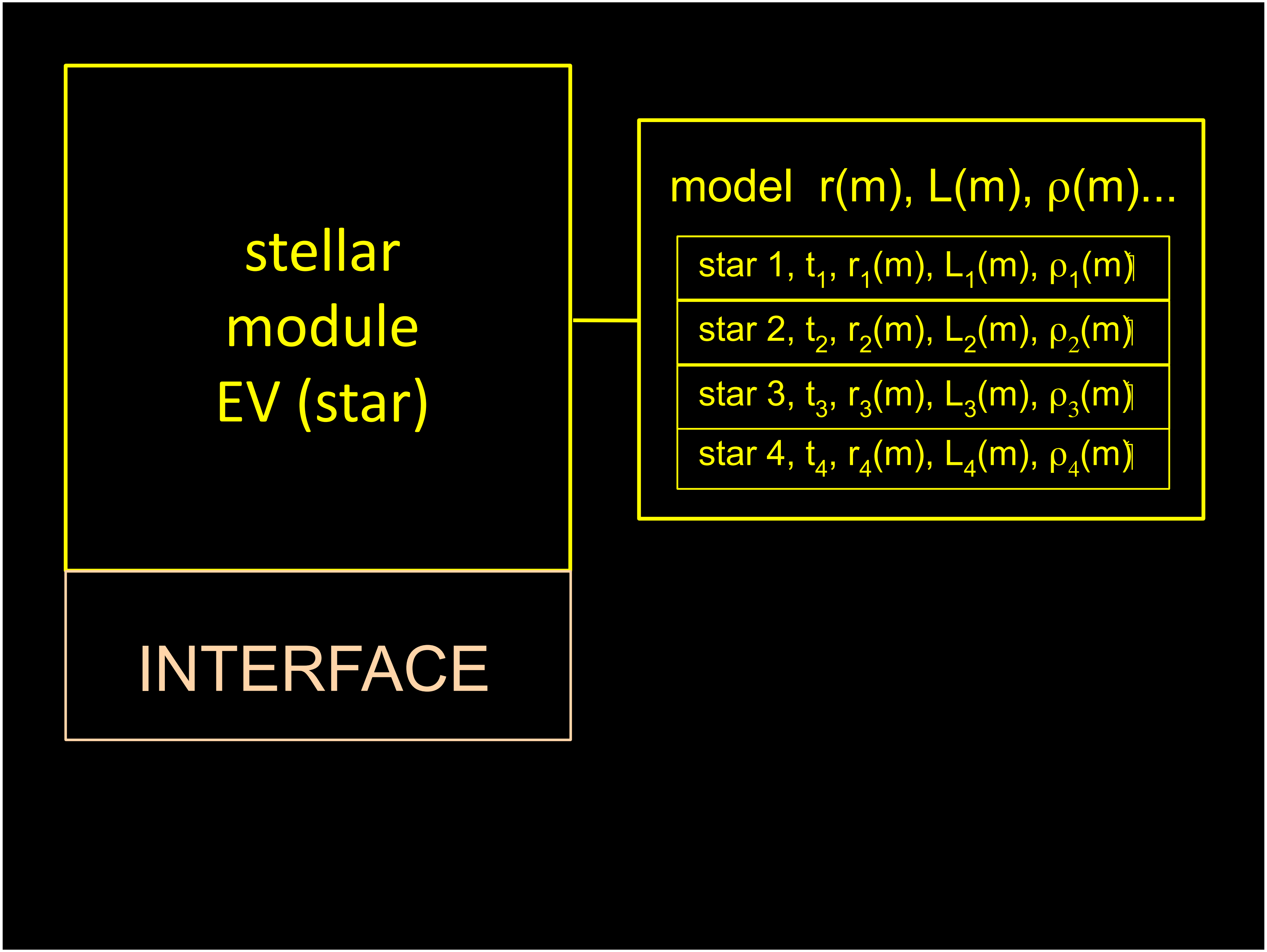

Star Module
toy model
analytic calculation
lookup table
heuristic recipe
full simulation
(real star...)
...
stellar data
INTERFACE
initialization
mass, composition
star ID
query
ID, time
mass
radius
temperature
(structure)
...
scheduling
ID
Δt

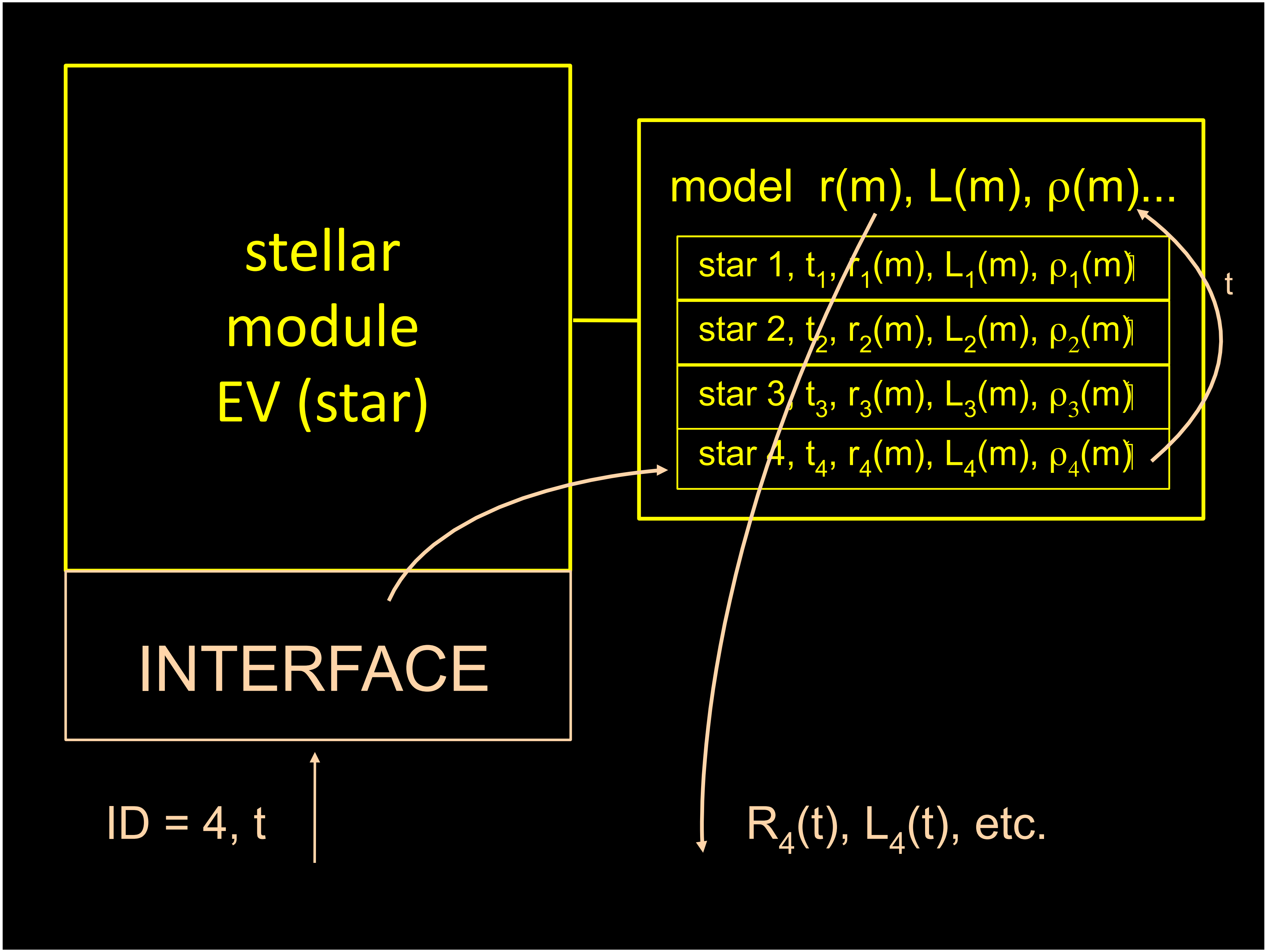


stellar
module
EFT89
INTERFACE
lookup R(M, t), L(M, t), ...
star 1, initial mass M_1
star 2, initial mass M_2
star 3, initial mass M_3
star 4, initial mass M_4



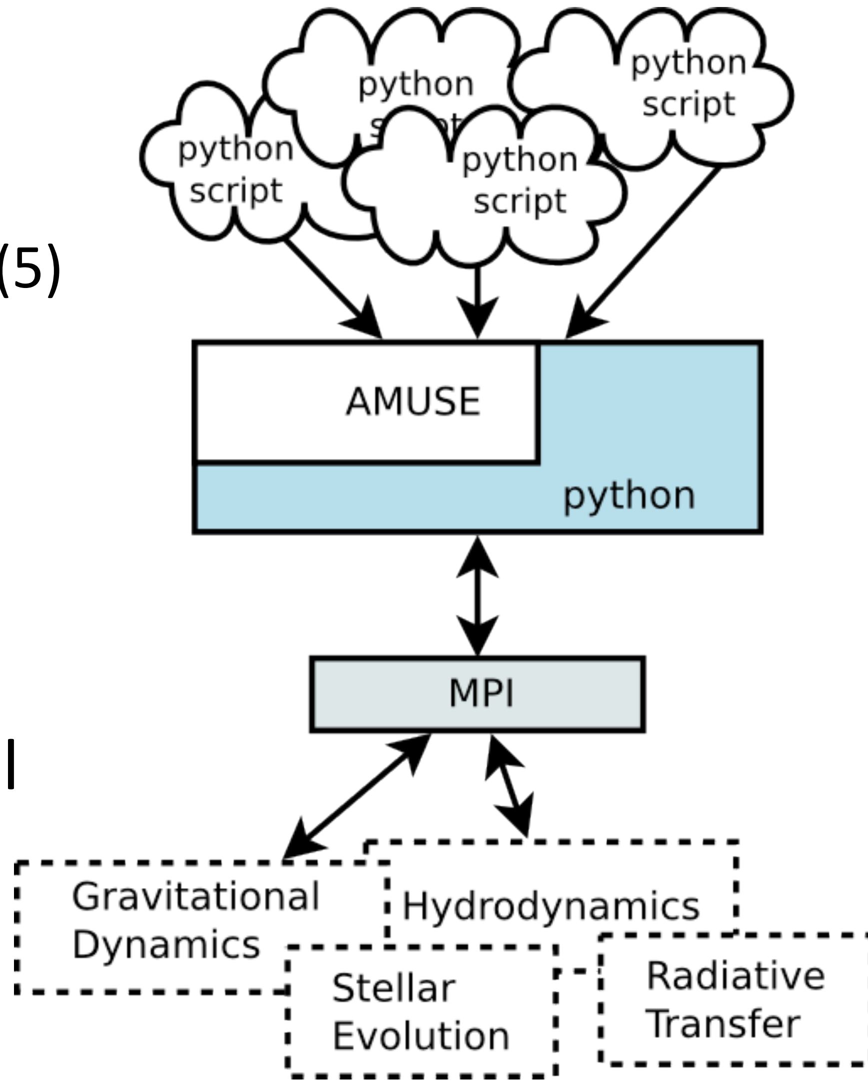
stellar
module
EFT89
INTERFACE
ID = 3, t
lookup R(M, t), L(M, t), ...
t
star 1, initial mass M_1
star 2, initial mass M_2
star 3, initial mass M_3
star 4, initial mass M_4
$R_3(t)$, $L_3(t)$, etc.


stellar
module
EV (star)
INTERFACE
model r(m), L(m), ρ(m)...
star 1, t1, r1(m), L1(m), ρ1(m)
star 2, t2, r2(m), L2(m), ρ2(m)
star 3, t3, r3(m), L3(m), ρ3(m)
star 4, t4, r4(m), L4(m), ρ4(m)

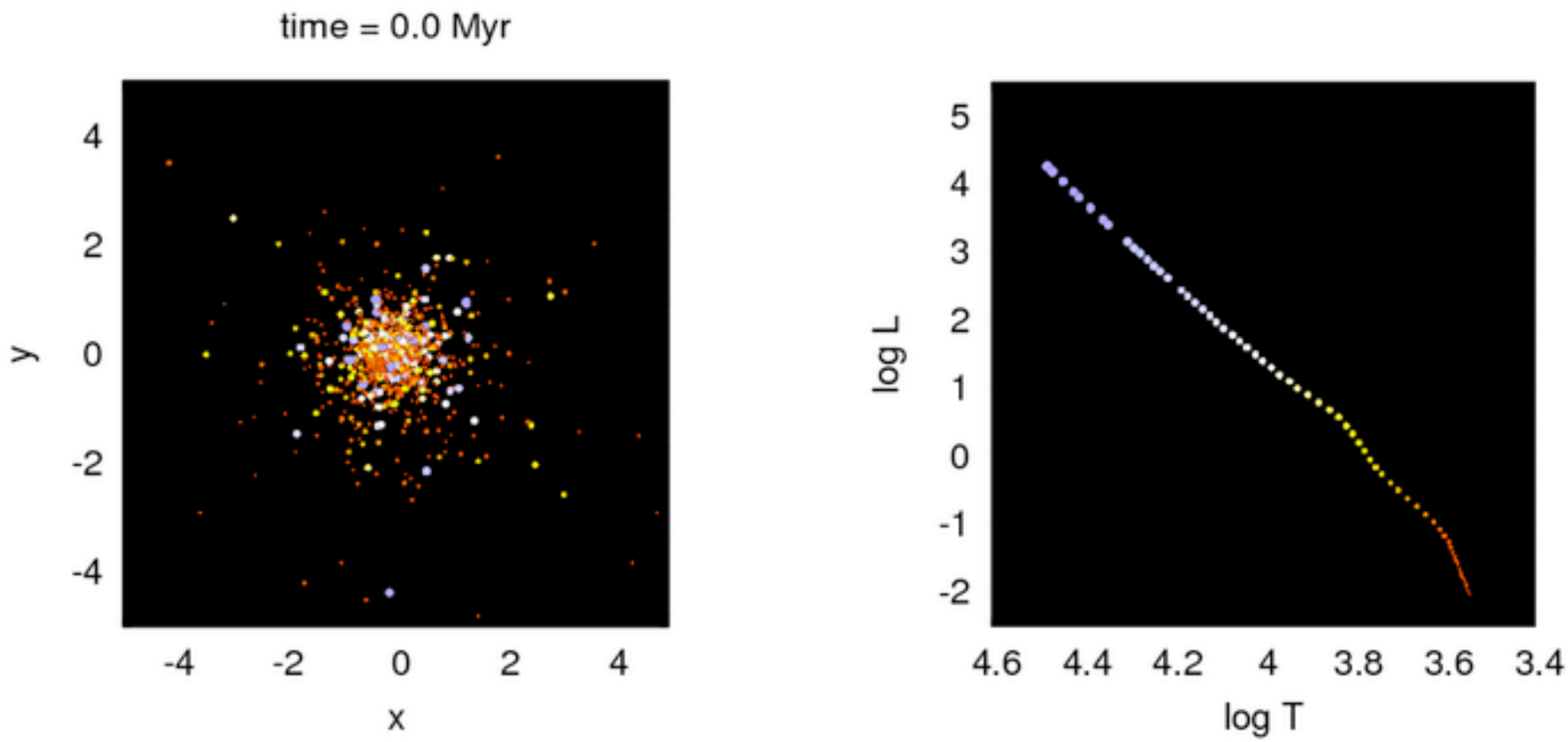

stellar
module
EV (star)
INTERFACE
ID = 4, t
model r(m), L(m), ρ(m)...
star 1, t1, r1(m), L1(m), ρ1(m)
star 2, t2, r2(m), L2(m), ρ2(m)
star 3, t3, r3(m), L3(m), ρ3(m)
star 4, t4, r4(m), L4(m), ρ4(m)
t
R4(t), L4(t), etc.

AMUSE Status

- currently have modules for
  - stellar dynamics (10)
  - stellar and binary evolution (5)
  - stellar collisions (2)
  - multiples (~1)
  - gas dynamics (4)
  - radiative transfer (3)
- all coupled via the top-level python layer



```
from amuse.community.ph4.interface import ph4 as gravity
from amuse.community.ph4.interface import EFT89 as stars
from amuse.community.ph4.interface import MMAS as coll

...   (initialization)

while time < end_max:

        time += dtime
        while gravity.get_time() < time:
                collision= gravity.evolve(time).check_coll()

                if collision > 0:
                        (id1,id2) = gravity.get_colliding_pair()
                        stars.evolve(gravity.get_time())
                        coll.collide_stellar_pair(id1, id2)

        stars.evolve(time)

print "end at t = ", time
```



time = 0.0 Myr
y
x
4
2
0
-2
-4
-4
-2
0
2
4
log L
log T
5
4
3
2
1
0
-1
-2
4.6
4.4
4.2
4
3.8
3.6
3.4