A new high-performance SPARC-SC code designed for realistic massive star cluster simulations

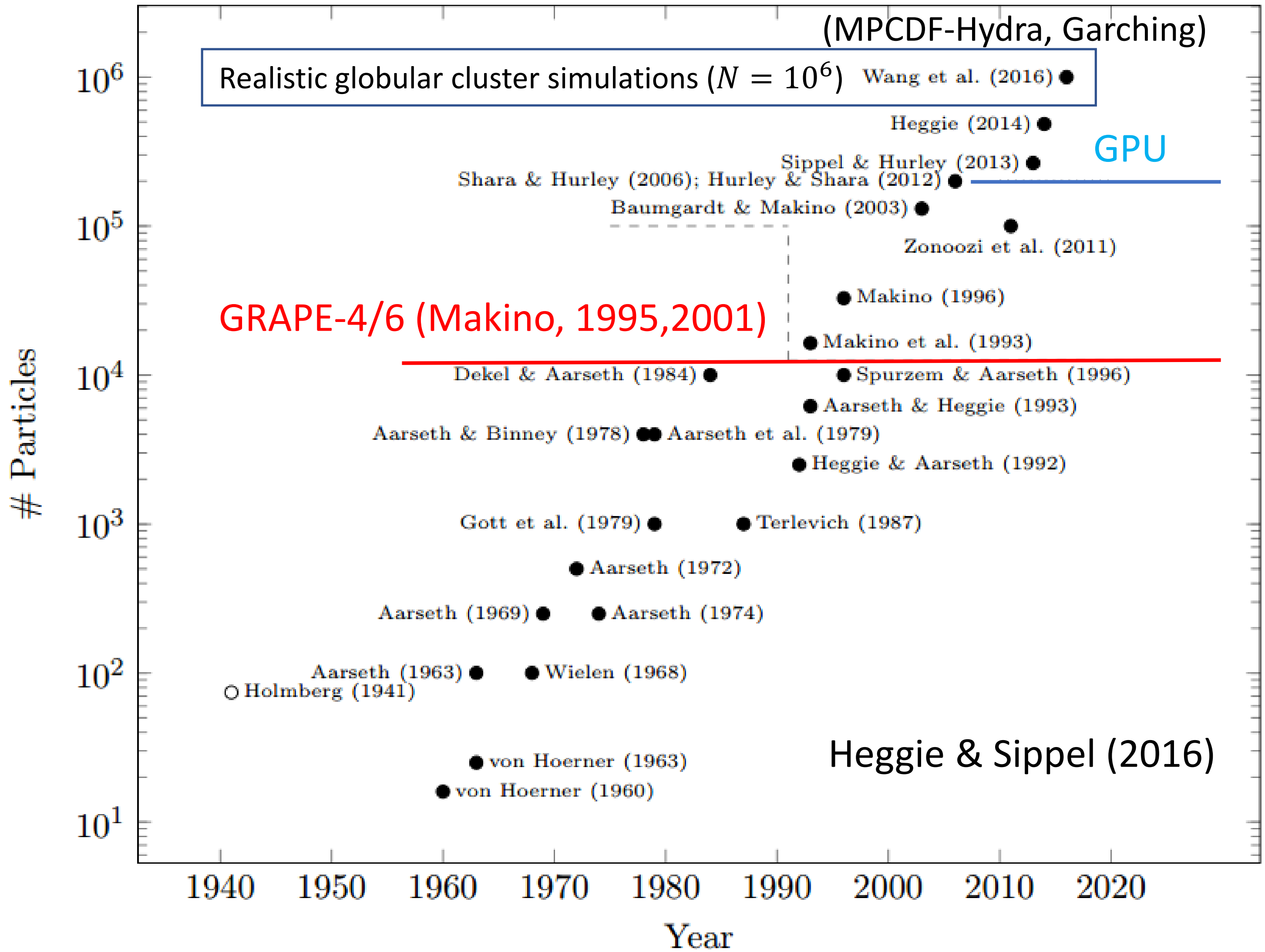
Long Wang (王龙) *Postdoctoral Researcher*

1. Argelander-Institut für Astronomie, Universität Bonn, Germany
2. RIKEN Advanced Institute for Computational Science, Japan

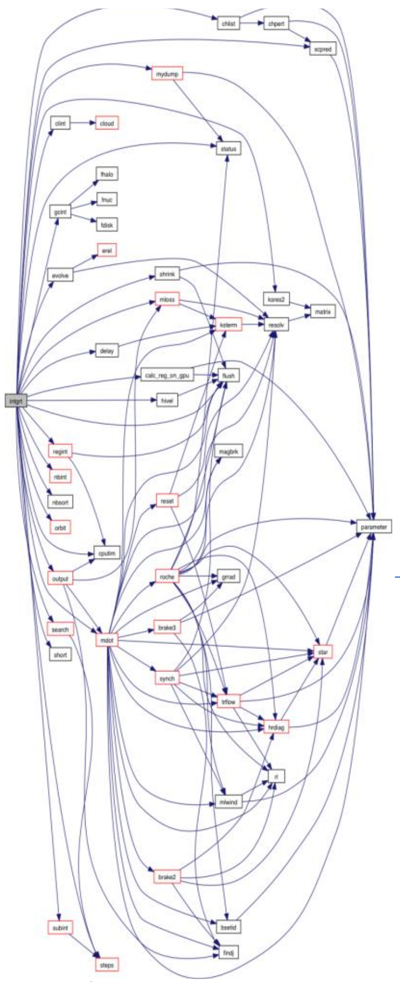
Collaborators:

- Jun Makino, Masaki Iwasawa, Keigo Nitadori (RIKEN-AICS)

Realistic star cluster simulations history

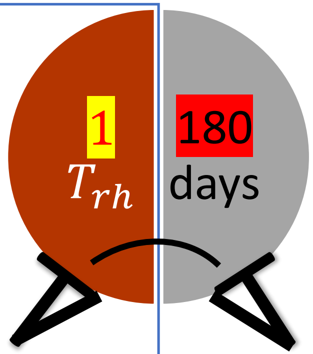


Current limit of million-body modelling



Computational time is too long

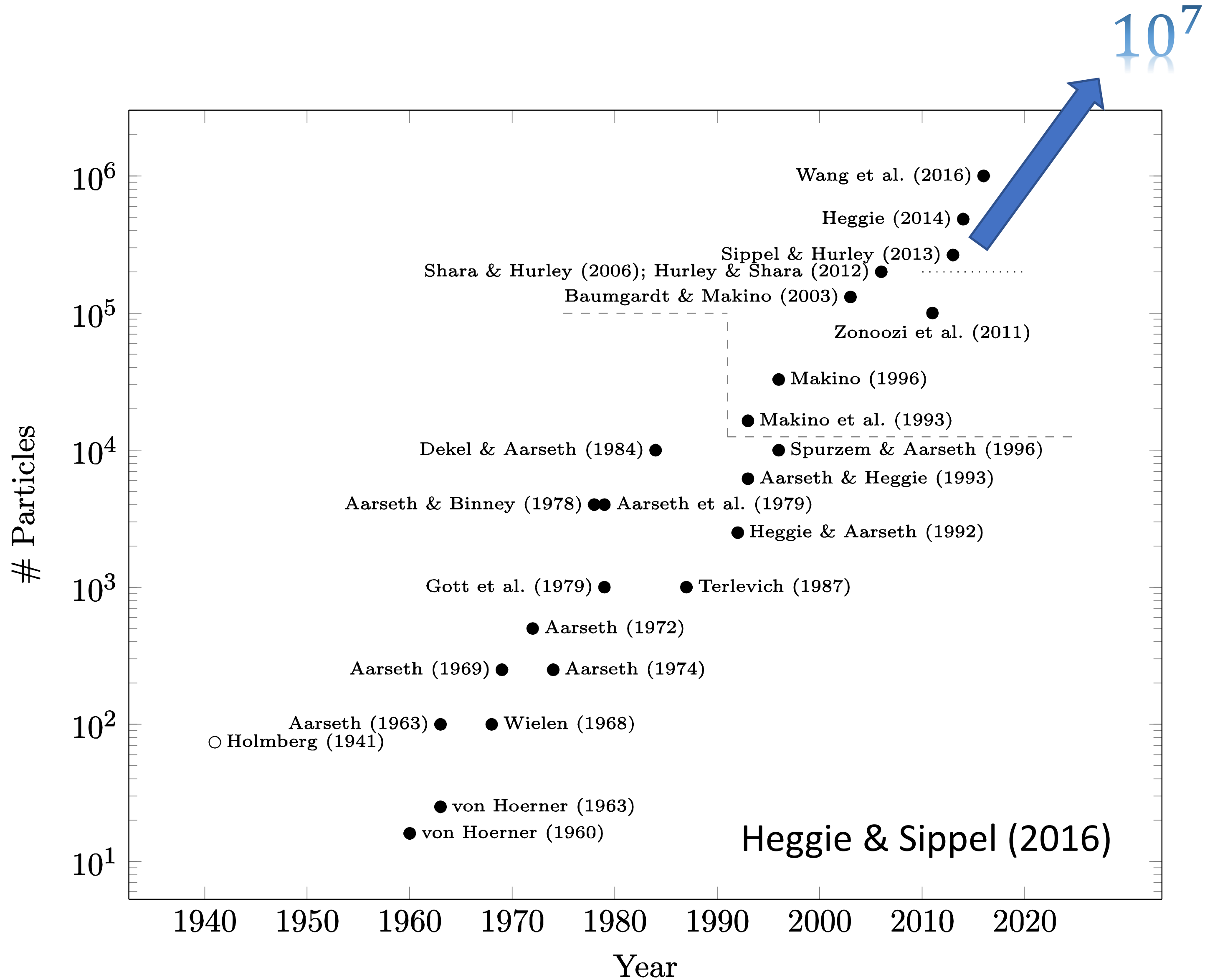
- Wang et al. (2016):
  - $N = 10^6$;
  - Primordial binary 5%
  - 160 CPU cores + 16 K20x GPUs



Difficult to improve the simulation code NBODY6++GPU

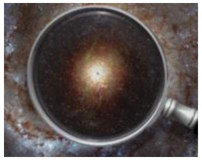
- Complexity programming style with Fortran 77.
- Data structure and algorithm limits

Towards $N \sim 10^7$



ω Cen

M60-UCD



Nuclear Star Clusters

Roughly speaking

Year	N	Method	Architecture
1960	30	Ind. Δt	Scalar
1970	100	Ind. Δt	Scalar
1980	300	Neighbor	Scalar
1990	3000	Neighbor	Vector
2000	30000	Ind. Δt	GRAPE
2010	100000	Neighbor	GPU
2020	3000000?	P^3T?	?

Architecture changes every 10 years.

Makino, NY, 2017

Optimization of N-body algorithm

NBODYX series (Aarseth)

Individual time step
(block time step)

$O(N\langle N_{act}\rangle)$

(Ahmad & Cohen 1973)
Neighbor scheme

$$O\left(\langle N_{Ne}N_{act}\rangle + \frac{N\langle N_{act}\rangle}{\alpha}\right)$$

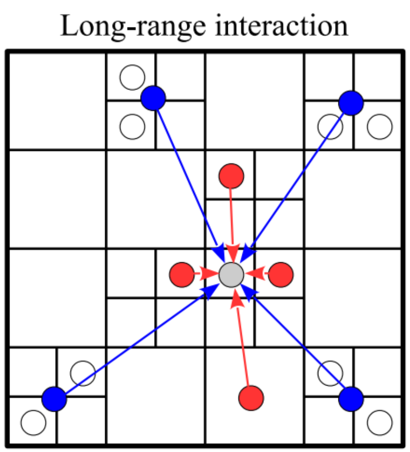
$O(N^2)$

Time

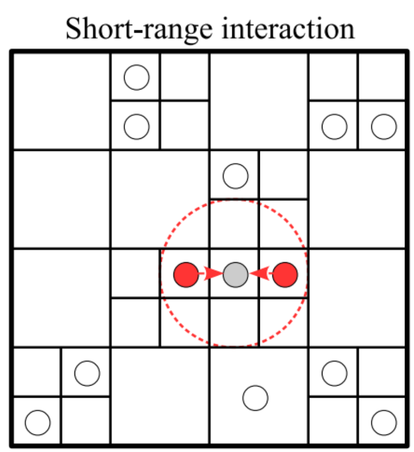
Particle tree

$O(N\log N)$

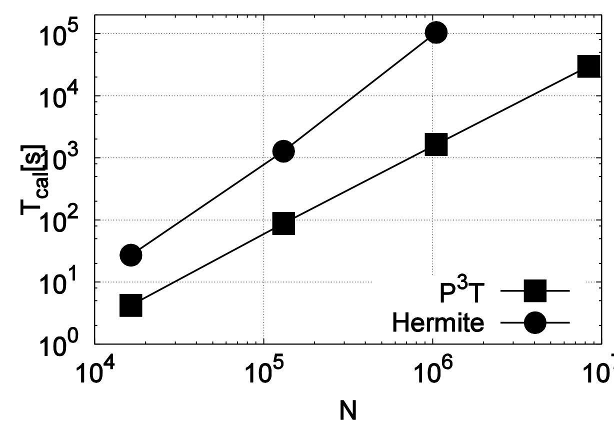
Space



Barnes-Hut tree



direct N-body



Iwasawa et al. (2016)

Particle Tree + Particle-Particle

Theoretical calculation cost scaling for $\mathrm{P}^3\mathrm{T}$

When core is large

- per crossing time: $N^{4/3} \log N$ ($N^{1/3}$ from timestep)
- per relaxation time $N^{7/3}$

When core is small: We need to make $O(N)$ hard binaries with triple interactions or binary-single star encounters

- Each hard binary requires constant cost, but with $\mathrm{P}^3\mathrm{T}$ this cost might be $N \log N$
- Total cost would be $N^2 \log N$

This means that even if we reduce the global tree step down to core crossing timescale, calculation cost is still N^2.

Makino, NY, 2017

Simulation turn-around time

When core is large:

- **For N crossing times, we need around $N^{4/3}$ steps. For $N = 10^7$, 10^9 steps.**
- **If we can make one timestep 1 msec (currently difficult... 10ms is doable), we can finish one run in 1 week or so.**

Small core scaling is still difficult to predict...

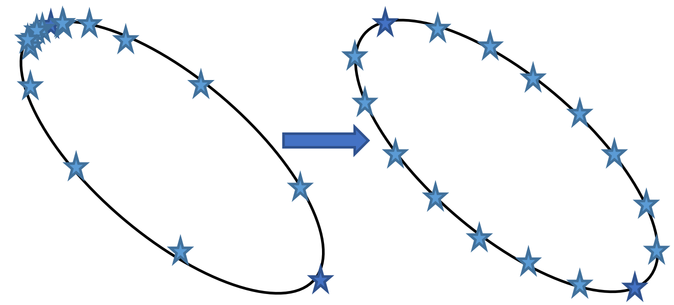
Makino, NY, 2017

Overcome timescale issue

Extreme timescale difference $T_{cr} \sim Myr$

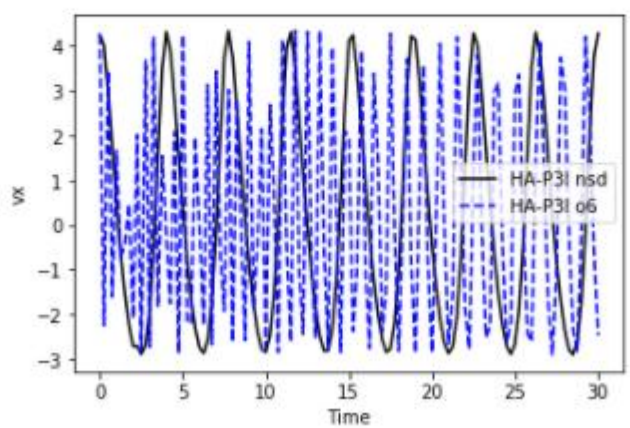
$T_{bin} \sim days$

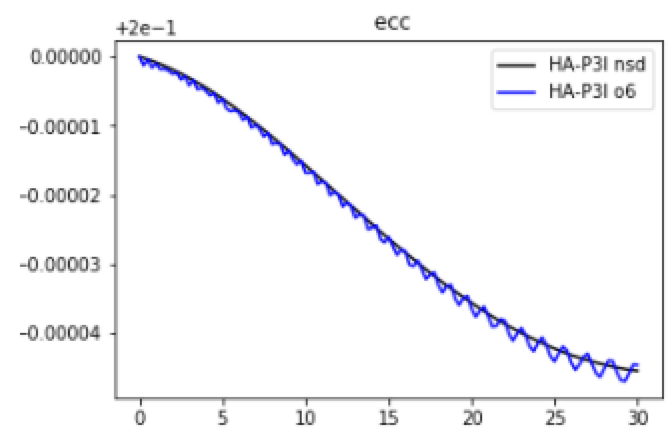
- Regularization
  - *Highly-eccentric binary & hyperbolic encounters*
    - KS $(t, \boldsymbol{r}, \boldsymbol{v})$
      - Kustaanheimo & Stiefel (1965)
      - Burden-Heggie (1967, 1968, 1973)
    - t only & Symplectic
      - AR (Mikkola & Tanikawa 1999)
      - Preto & Tremaine (1999)



Symplectic leapfrog

- Slowdown (Mikkola &Aarseth 1996)
  - *Reduce timescale gap.* $H_{SD} = \kappa^{-1} H_{bin} + (H - H_{bin})$





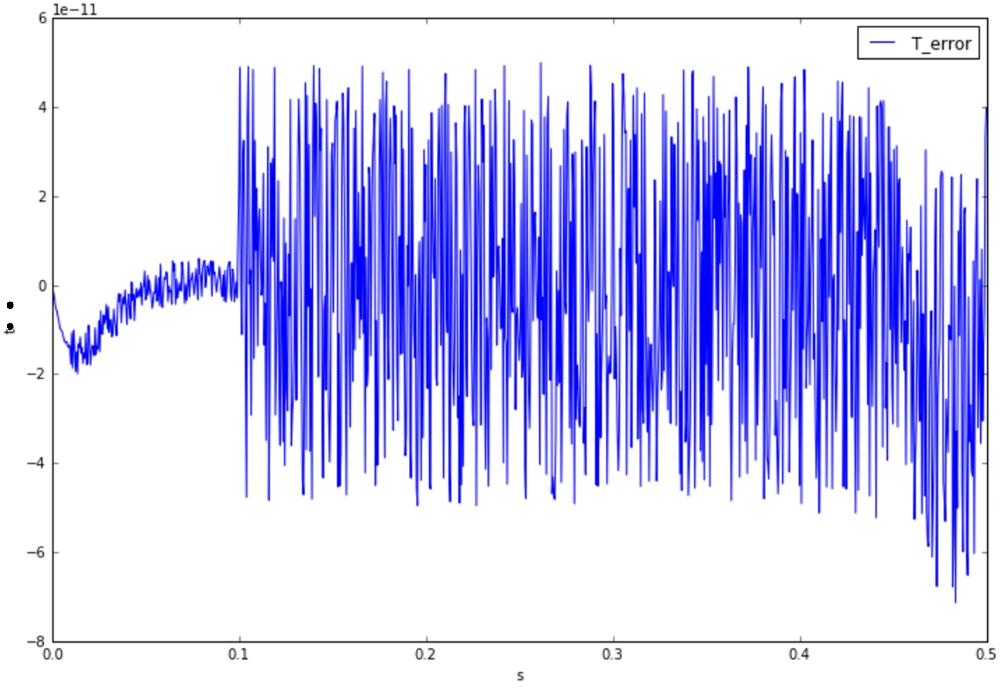
Secular motion is correctly; phase information is lost

Two issues for combination of AR & P^3T

- Time synchronization for AR method
  - time in AR integration is unknown before integration
- How to deal with perturbations
  - strong perturbation
  - weak perturbation

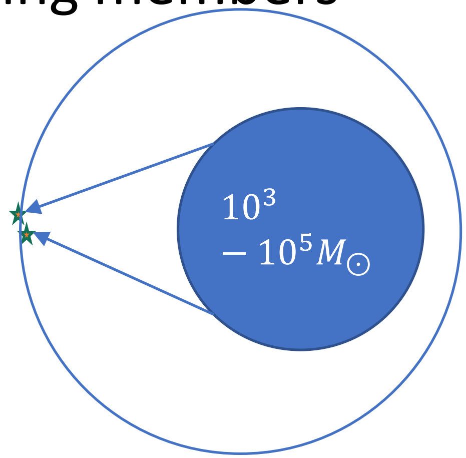
Time synchronization for AR method

- AR method
  - Symplectic Leap-frog + B.S. integrator
  - $T_{AR} \rightarrow T_{global}$
- Dense Output
  - (Hairer & Ostermann, 1990).
  - Interpolation function: T(s)

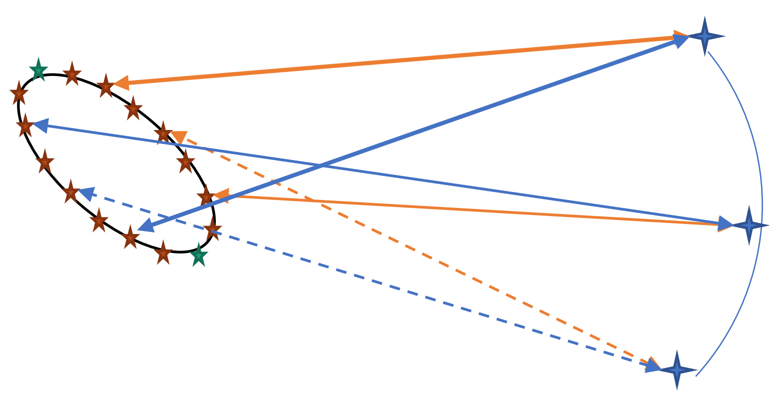


Interaction between binaries and star clusters

- **Strong perturbation** (AMUSE, NBODY6, Monte-Carlo)
  - Few-body interactions
  - Binary formation, disruption and exchanging members
- **Weak perturbation** from distant stars
  - NBODY6: tidal cutoff:
    - $r < \left[\frac{2m_j}{m_{cm}\gamma_{min}}\right]^{1/3} R_{max}$
  - Eccentricity evolution



- **Random phase issue** due to extremely different time steps

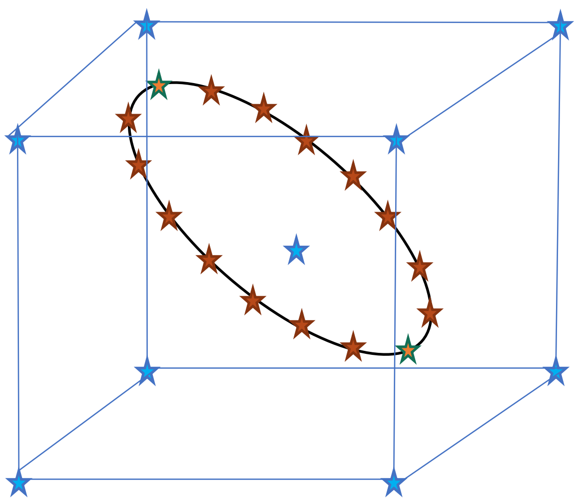


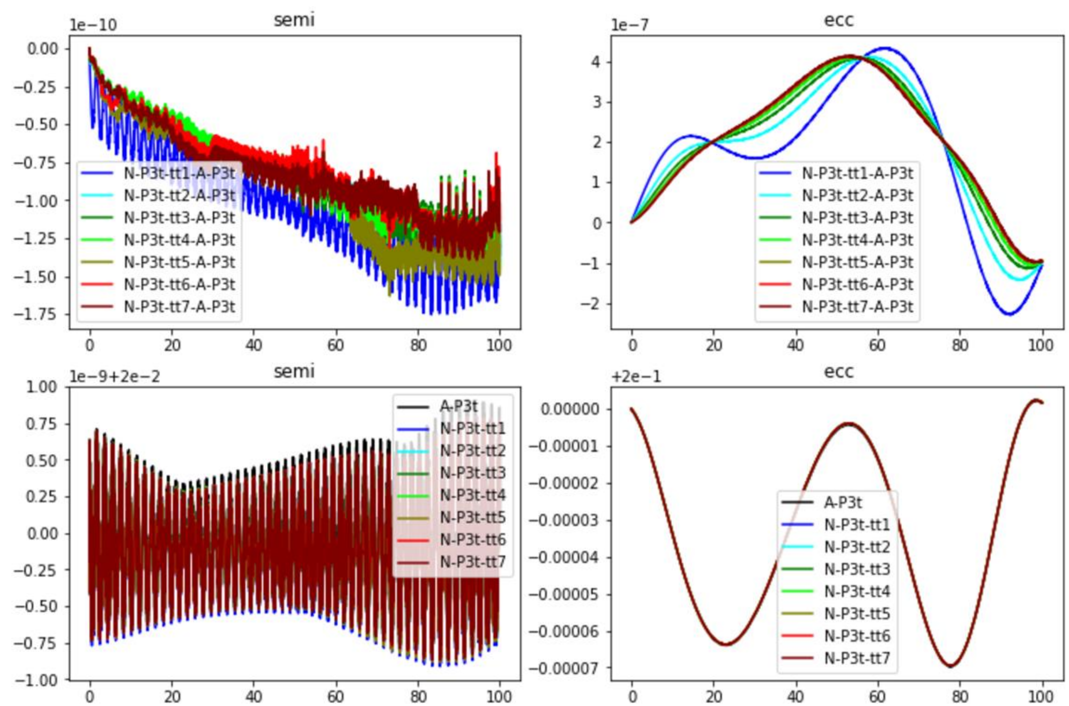
Tidal-tensor perturbation scheme

$$\boldsymbol{F}(\boldsymbol{r}) = \boldsymbol{F}(\boldsymbol{r}_{cm}) + \sum_i \frac{\partial \boldsymbol{F}}{\partial r_i}(r_i - r_{cm,i}) + \frac{1}{2}\sum_i \sum_j \frac{\partial^2 \boldsymbol{F}}{\partial r_i \partial r_j}(r_i - r_{cm,i})(r_j - r_{cm,j})$$

Create artificial sample points to obtain

1. *local tidal tensor*
2. *averaged counter-force*



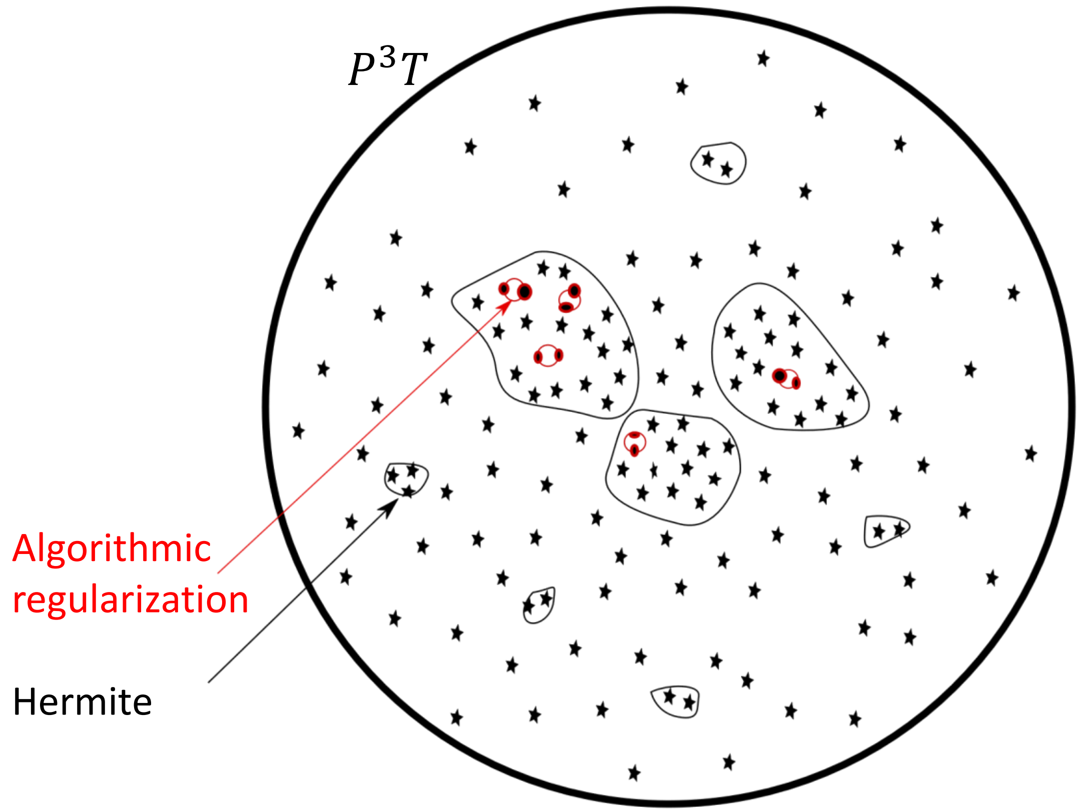


2^{nd} order : 16 unknown parameters

8 Points : 24 measured force information

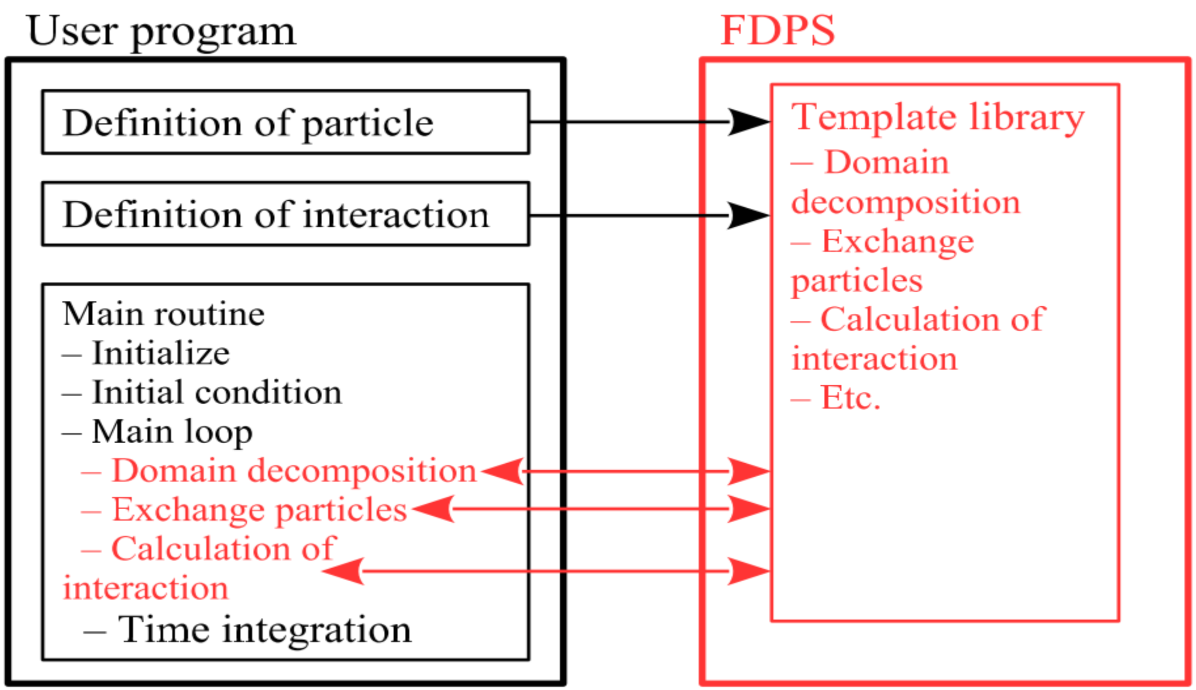
Symplectic Particle tree & Algorithmic Regularization Code for Star Clusters (SPARC-*SC*)

https://github.com/AICS-SC-GROUP/SPARC-SC (experimental version)
https://github.com/lwang-astro/TSARC
(Algorithmic Regularization Chain implemented in C++ template library)



FDPS A framework for developing parallel particle simulation codes (Iwasawa et al. 2016)

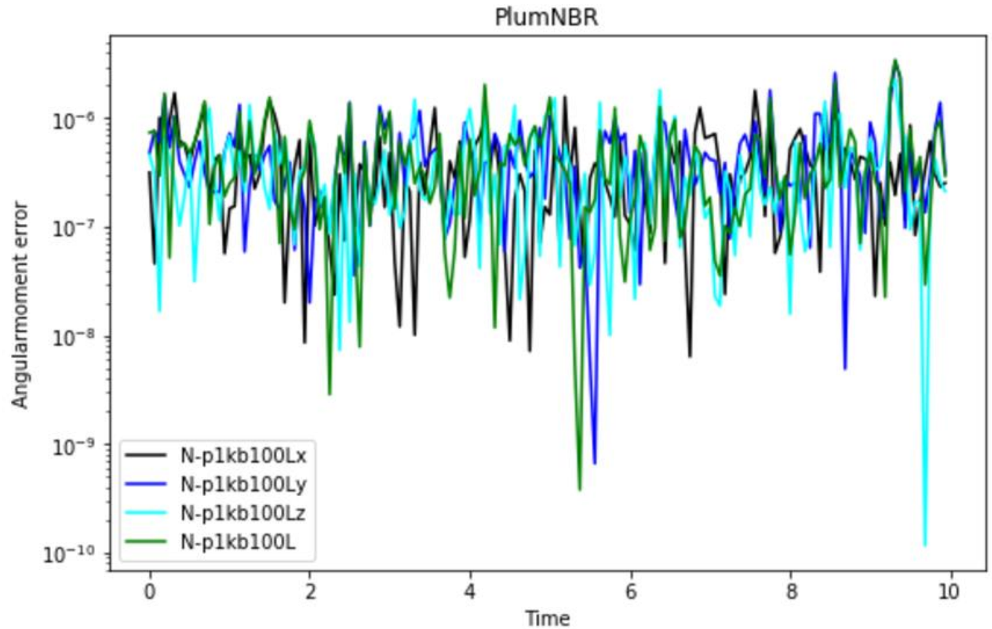
- Nature Scientists (Users)
  - Pair interaction function
  - Integration method
- Computational scientists (FDPS developers)
  - Parallelization
  - Deep optimization

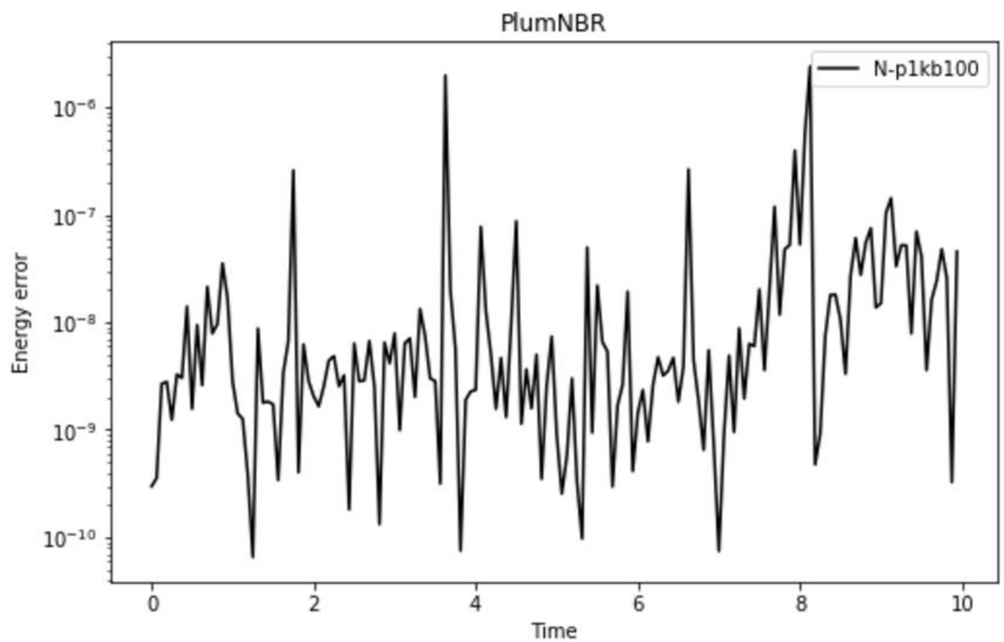


Test case

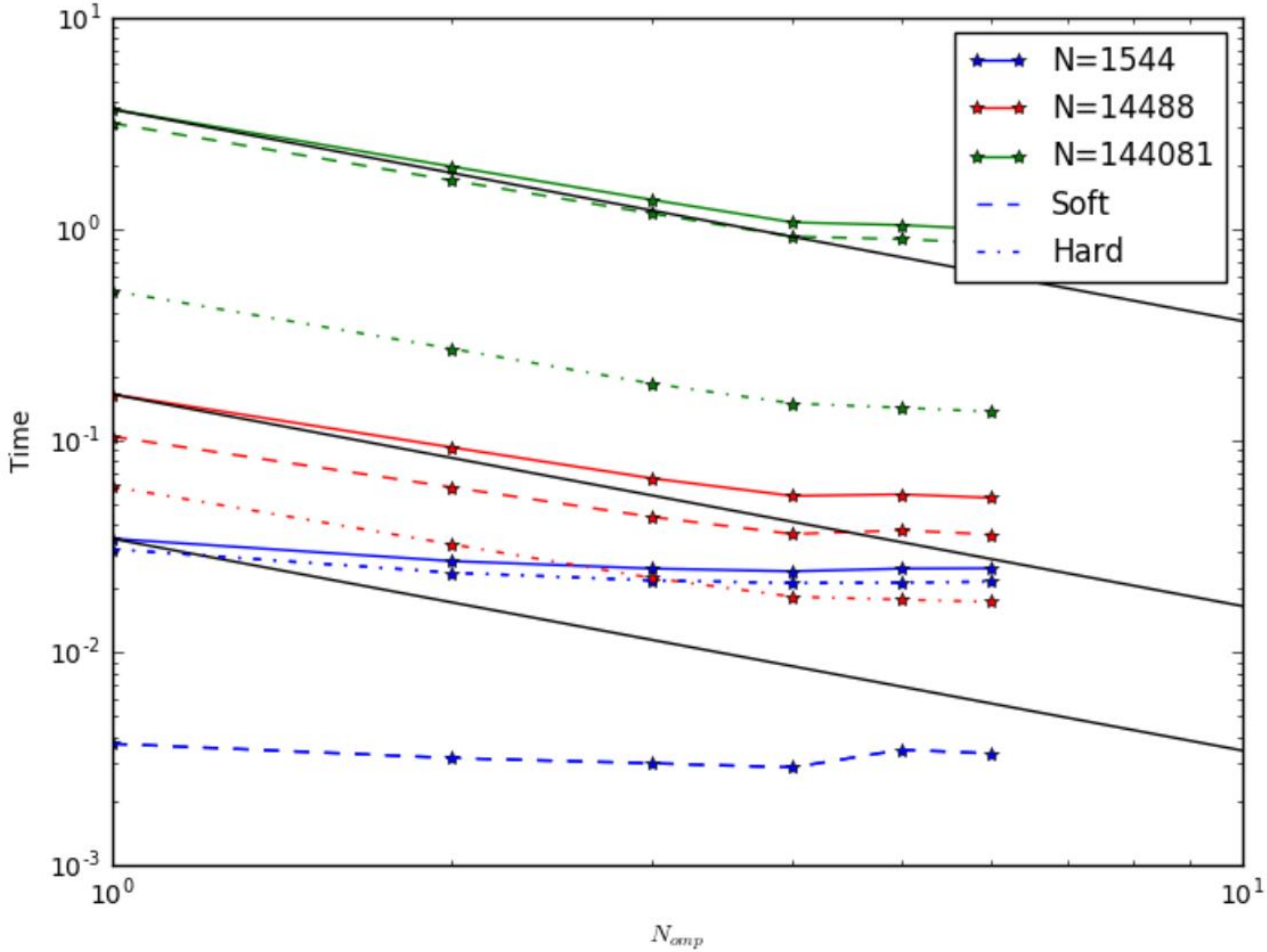
- $N = 1000$
- $N_{bin} = 100$
- Kroupa (2001) IMF

- Support Platform
  - GRAPE
  - SPARC64
    - K-computer
  - X86_64
    - Intel CPU
    - Xeon-Phi
  - GPU
    - Nvidia (CUDA)





Scaling with OpenMP threads & N

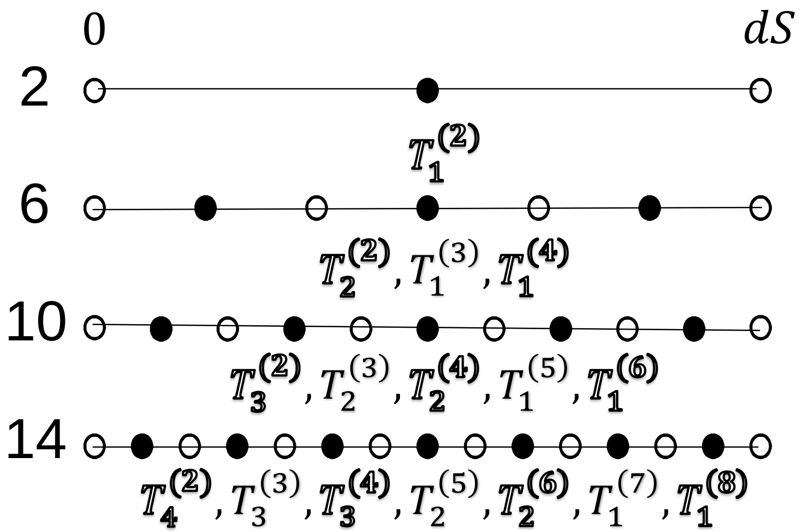


Future work

- Improve the performance of AR
  - B.S. is relative time-consuming compared with P^3T
  - Or use special Hardware like *Intel Xeon Phi*
- Add API for stellar evolution recipes
- Implement Post-Newtonian gravity to AR
  - Modified middle point integrator (double Leap-frog)

Extrapolation with Dense Ouput

Hairer & Ostermann (1990) (4k): n_j =(2, 6, 10, 14 ...)



Extrapolation

- $T_{1,1}$
- $T_{2,1} \rightarrow T_{2,2}$
- $T_{3,1} \rightarrow T_{3,2} \rightarrow T_{3,3}$
- $T_{4,1} \rightarrow T_{4,2} \rightarrow T_{4,3} \rightarrow T_{4,4}$

1. High order derivate: $T_k^{(i)} = \frac{\delta^{i-1} T_1^{(1)}}{(2\mathrm{ds_j})^{\mathrm{i}-1}}$; $T_1^{(1)}$ is known during integration; $ds_j = \frac{dS}{n_j}$
2. Extrapolating $T_k^{(i)}$ $(k = 1, \kappa)$ to high accurate: $T^{(i)} = T^{i}_{\kappa,\kappa}$; $i_{max} = 2j - 1$
3. Polynomial interpolation function $T_p(s)$ using $T(0), T(dS), T^{(i)}$

Summary

- We develop a high-performance Symplectic Particle tree & Algorithmic Regularization simulation Code for Star Clusters (SPARC-SC)
- Aimed at:
  - Speed up million-body globular cluster simulations
  - $\geq 10^7$ collisional systems with binaries