

SCF-FDPS: A Fast N-Body Code for Simulating Disk-Halo Systems

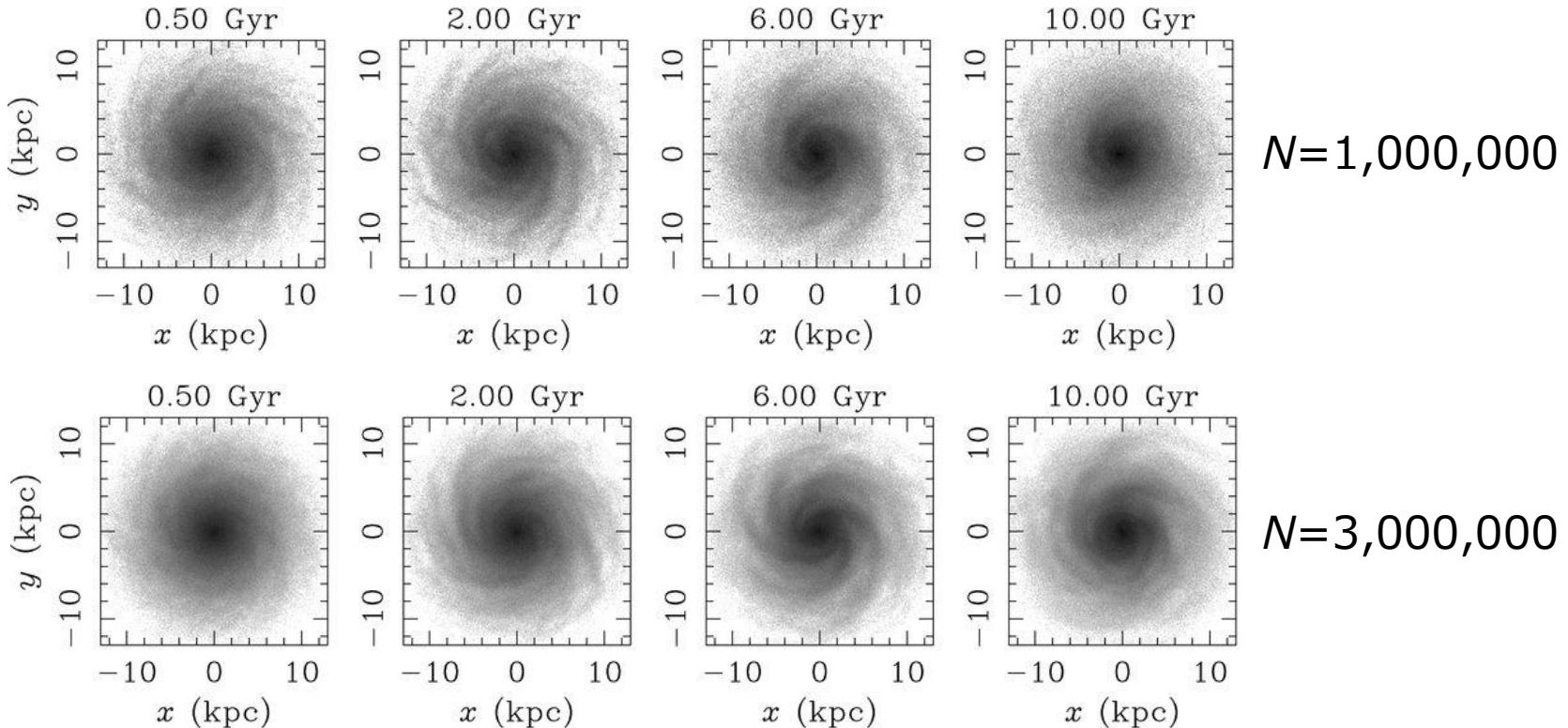
published in ApJ, 948: 29, 2023

穂積俊輔（滋賀大学）, 似鳥啓吾（理化学研究所）, 岩澤全規（松江高専）

1. 円盤銀河の N 体シミュレーション

大量の円盤粒子の必要性

Disk simulations embedded in a fixed halo



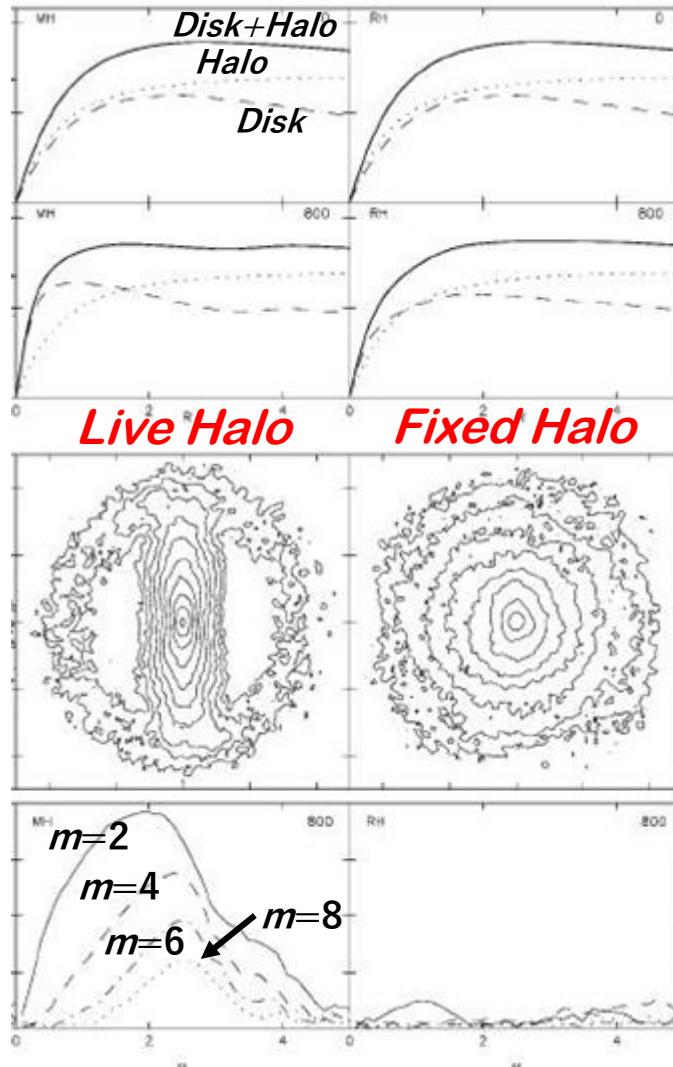
$N=1,000,000$

$N=3,000,000$

◎円盤は回転で支配された“冷たい”系
⇒Poissonノイズによる円盤加熱の回避
⇒大量の円盤粒子

自己重力ハローの必要性

Massive Halo (MH) Rigid Halo (RH)



V_{rot} at $t=0$

V_{rot} at $t=800$

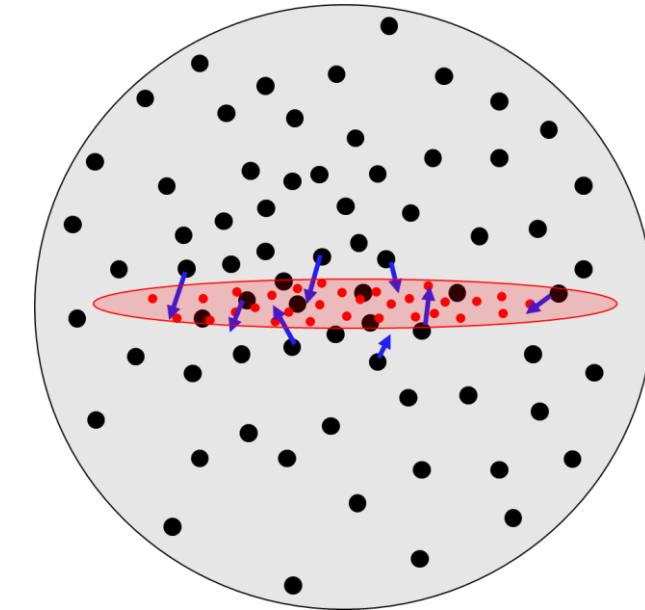
Density Contours
at $t=800$

Fourier Amplitudes
at $t=800$

◎バー・パターンとハロー粒子との共鳴
⇒大振幅バーの励起
⇒ハローは外場ではなく、自己重力系として扱う必要性

大量のハロー粒子の必要性

- ・自己重力ハロー
 - ・ハロー粒子が円盤通過時のショットノイズの回避
⇒ハロー粒子と円盤粒子は等質量
 - ・ハローは円盤の~10倍の質量
- ⇒大量のハロー粒子が必要



ハロー粒子の質量が円盤粒子の質量よりも大きい場合に生じるハロー粒子が円盤通過時に引き起こすショットノイズ

Ex. 円盤粒子1億体に対してハロー粒子10億体程度

高速な*N*体シミュレーションコードの必要性

2. 高速 N 体シミュレーションコード

N 体シミュレーションコードの高速化

- ツリー法

- 近傍の粒子は2体間の重力を計算し、遠方粒子は設定した見込み角度 θ 内にある粒子をまとめて1個の粒子のように扱って重力を計算する方法
- 重力の精度は見込み角 (opening angle) θ に依存
⇒ θ が小さいほど重力は高精度に計算される
- 並列計算用のフレームワーク
⇒ *FDPS* (Framework for Developing Particle Simulators)による極限的なチューニング
(*Iwasawa et al.*, 2016, PASJ, 68, 54; *Namekata et al.*, 2018, PASJ, 70, 70)
⇒ これ以上の高速化は非常に困難

SCF (Self-Consistent Field) 法の組み込み

SCF法の概要

References:

Clutton-Brock, 1973, Ap&SS, **23**, 55

Hernquist & Ostriker, 1992, ApJ, **386**, 375

系の密度とポテンシャル(ρ, Φ)を直交基底関数系(ρ_{nlm}, Φ_{nlm})で展開して
Poisson方程式 $\nabla^2\Phi(\mathbf{r}) = 4\pi G\rho(\mathbf{r})$ を解く方法

- 直交基底関数系: (ρ_{nlm}, Φ_{nlm})

- Poisson方程式: $\nabla^2\Phi_{nlm}(\mathbf{r}) = 4\pi G\rho_{nlm}(\mathbf{r})$

- biorthogonality: $\int \rho_{nlm}(\mathbf{r})[\Phi_{n'l'm'}(\mathbf{r})]^* d\mathbf{r} = \delta_{nn'}\delta_{ll'}\delta_{mm'}$

n : radial expansion
 l, m : angular expansions

$$\rightarrow \begin{cases} \rho(\mathbf{r}) = \sum_{nlm} A_{nlm} \rho_{nlm}(\mathbf{r}) \\ \Phi(\mathbf{r}) = \sum_{nlm} A_{nlm} \Phi_{nlm}(\mathbf{r}) \end{cases}$$

展開係数: $A_{nlm} = \int \rho(\mathbf{r})[\Phi_{nlm}(\mathbf{r})]^* d\mathbf{r}$

- N 粒子系: $\rho(r, \theta, \phi) = \sum_{k=1}^N m_k \frac{1}{r_k^2} \delta(r - r_k) \delta(\phi - \phi_k) \delta(\cos \theta - \cos \theta_k)$

$$A_{nlm} = \sum_{k=1}^N m_k [\Phi_{nlm}(\mathbf{r}_k)]^* \rightarrow \mathbf{a}(\mathbf{r}) = -\nabla \Phi(\mathbf{r}) = -\sum_{nlm} A_{nlm} \nabla \Phi_{nlm}(\mathbf{r})$$

→ 重力場 (*Field Method*)

SCF法によるシミュレーション

Basis Set (ρ_{nlm} , Φ_{nlm}):

$$\nabla^2 \Phi_{nlm}(\mathbf{r}) = 4\pi G \rho_{nlm}(\mathbf{r})$$

$$\int \rho_{nlm}(\mathbf{r}) [\Phi_{n'l'm'}(\mathbf{r})]^* d\mathbf{r} = \delta_{nn'} \delta_{ll'} \delta_{mm'}$$

n : radial expansion
 l, m : angular expansions

$$\nabla^2 \Phi(\mathbf{r}) = 4\pi G \rho(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_{nlm} A_{nlm} \rho_{nlm}(\mathbf{r})$$

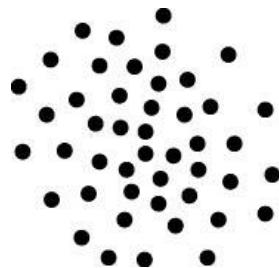
$$\Phi(\mathbf{r}) = \sum_{nlm} A_{nlm} \Phi_{nlm}(\mathbf{r})$$

References:

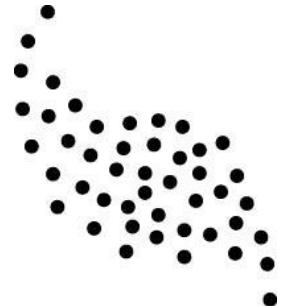
Clutton-Brock, 1973, Ap&SS, 23, 55

Hernquist & Ostriker, 1992, ApJ, 386, 375

Particle Distribution at t



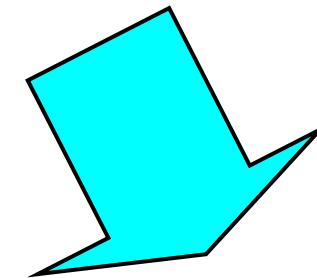
Multiply ρ by Φ_{nlm}



Obtain Expansion Coefficients A_{nlm}

$$A_{nlm} = \int \rho(\mathbf{r}) [\Phi_{nlm}(\mathbf{r})]^* d\mathbf{r}$$

$$= \sum_{k=1}^N m_k [\Phi_{nlm}(\mathbf{r}_k)]^*$$



SCF simulation procedure

Particle Distribution at $t + \Delta t$

Suitable Integration Scheme

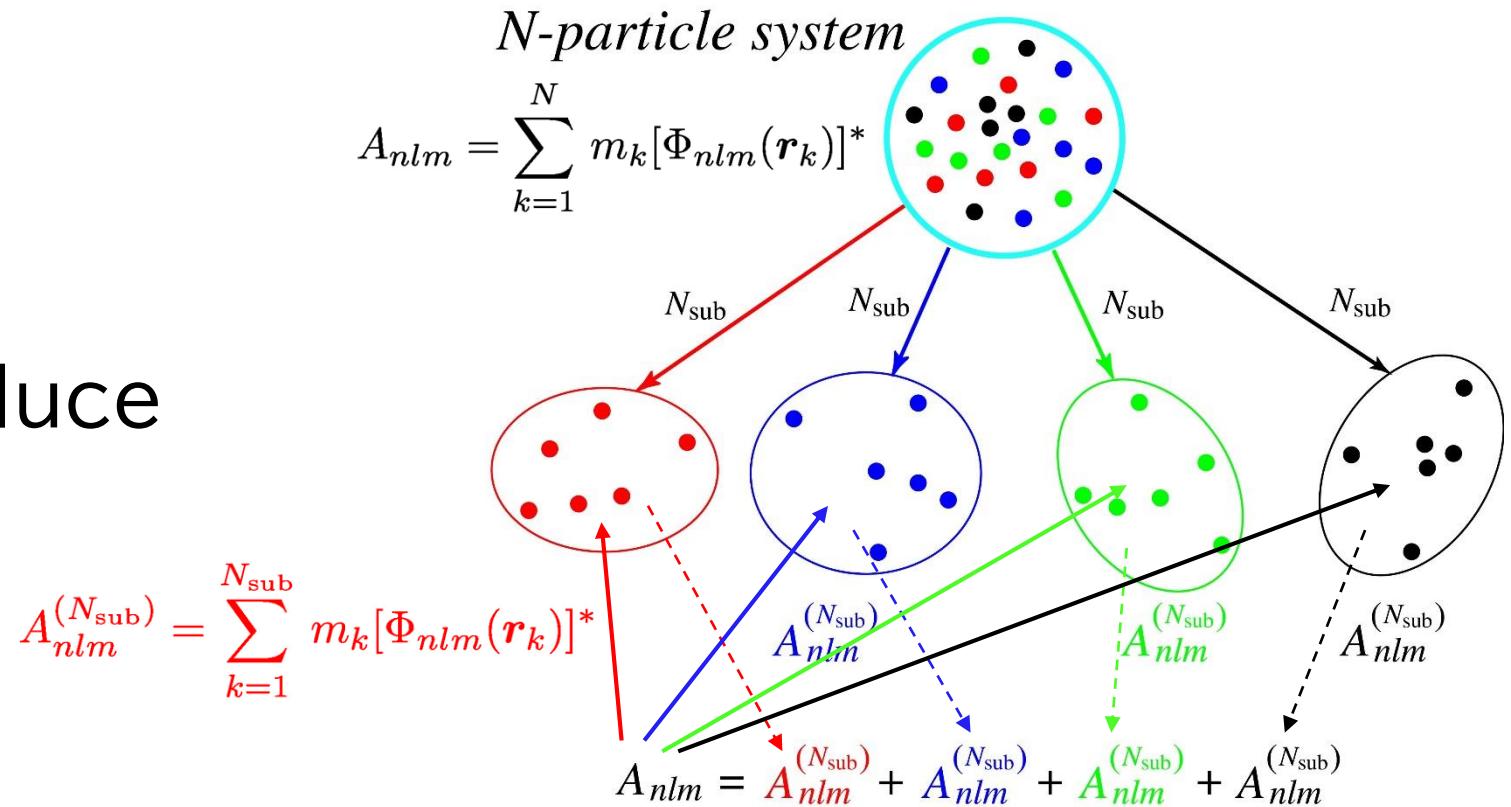
Compute Accelerations $\mathbf{a}(\mathbf{r})$

$$\mathbf{a}(\mathbf{r}) = - \sum_{nlm} A_{nlm} \nabla \Phi_{nlm}(\mathbf{r})$$

$\nabla \Phi_{nlm}(\mathbf{r})$ can be analytically calculated in advance, once the basis set is specified.

SCF法の特徴

- ・完全な並列性
- ・通信はMPI_Allreduce のみでよい
- ・計算時間 $\propto N \times (n_{\max}+1) \times (l_{\max}+1) \times (m_{\max}+1)$

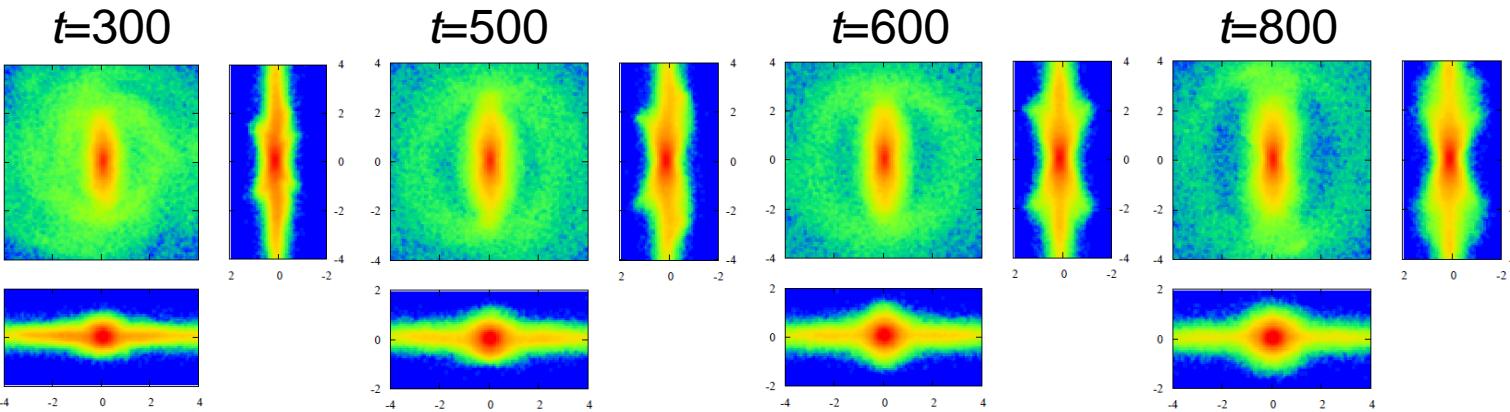


Application of SCF Method: Simulation of a Disk-Halo System

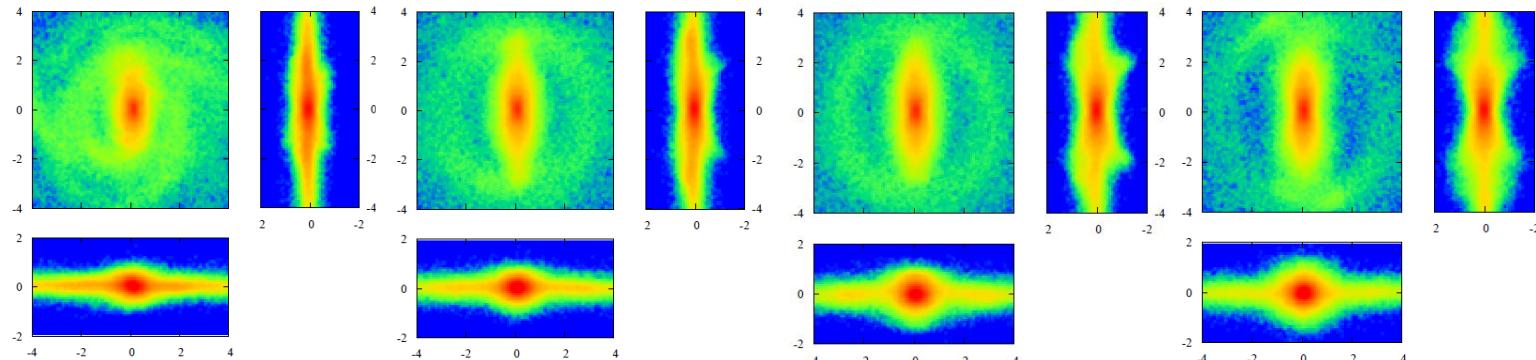
● exponential disk embedded in a live NFW halo

● $N_{\text{disk}}=200\,192$, $N_{\text{halo}}=1\,000\,960$

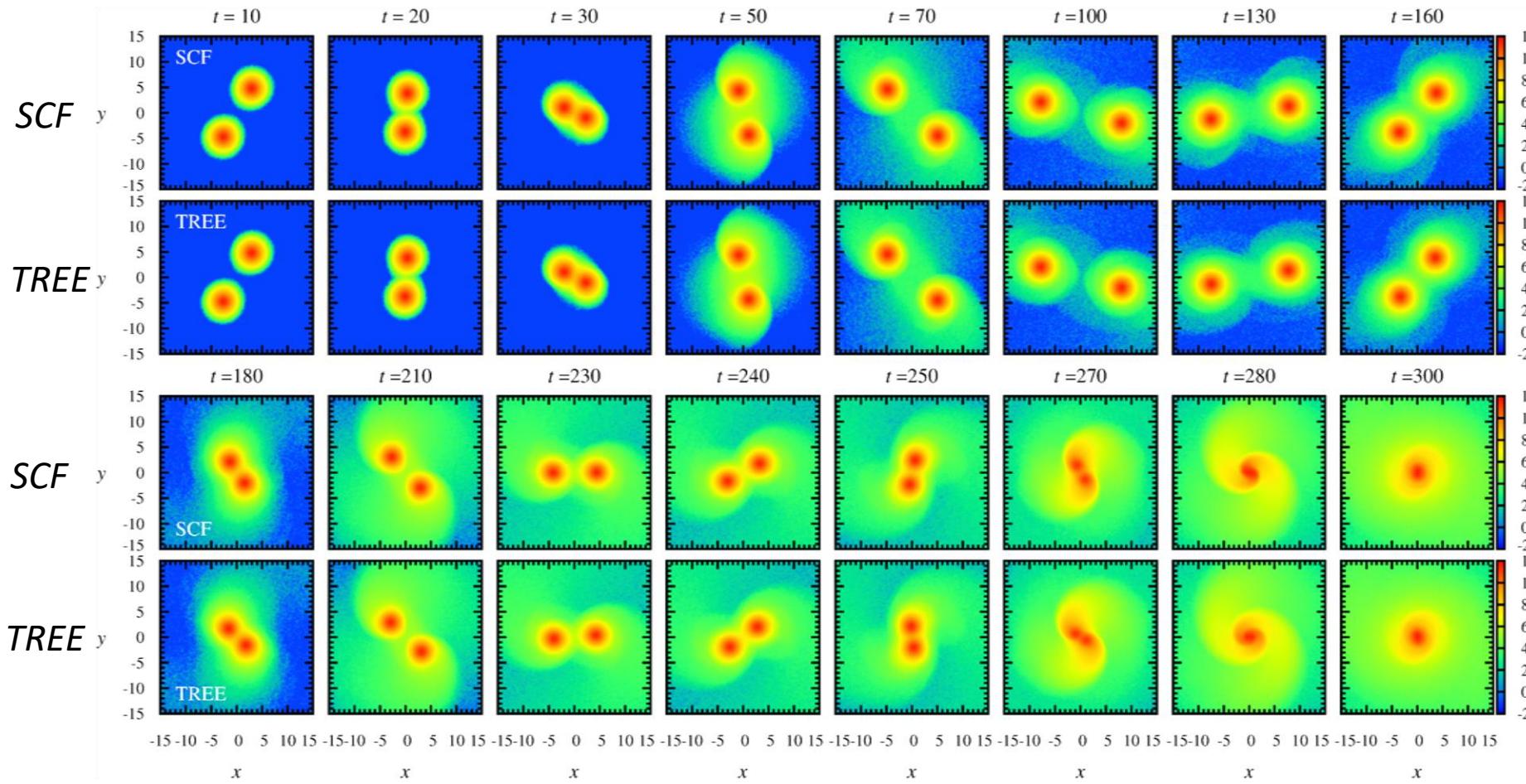
SCF:
 $n_{d,\text{max}}=64$
 $m_{d,\text{max}}=24$
 $l_{d,\text{max}}=24$
 $n_{h,\text{max}}=32$
 $l_{h,\text{max}}=12$
 $m_{h,\text{max}}=12$



TREE:
 $\varepsilon=0.04$
 $\theta=0.5$



Application of SCF Method: Merging Simulation of Two King Models

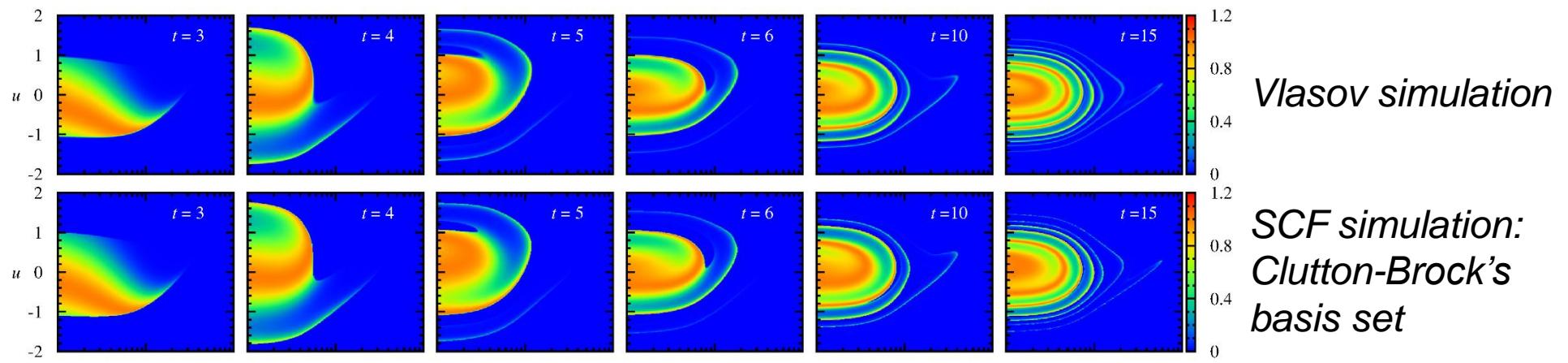


- Density evolution of Systems 1 and 2 for impact parameter = 10
- Interaction forces are calculated by expanding the density and potential with respect to the center of mass of the total system

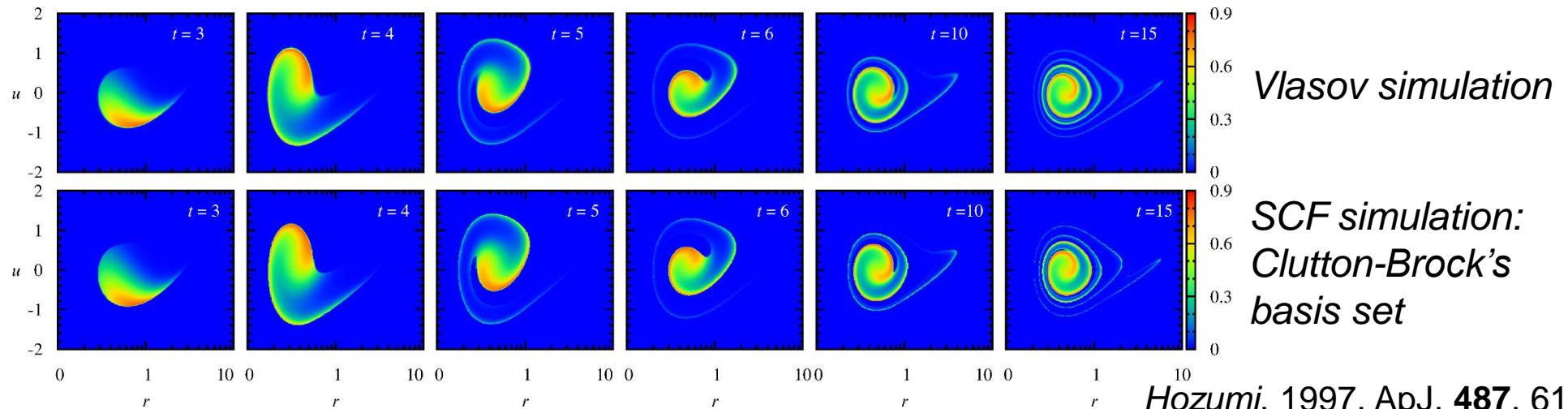
Application of SCF Method: Phase-Space Evolution of a Collapsing Uniform Sphere

● Symmetric collapse with $|2T/W| = 1/2$, Vlasov : $(N_r, N_u, N_j) = (400, 401, 50)$, SCF : $N = 10\,000\,000$, $n_{max} = 64$

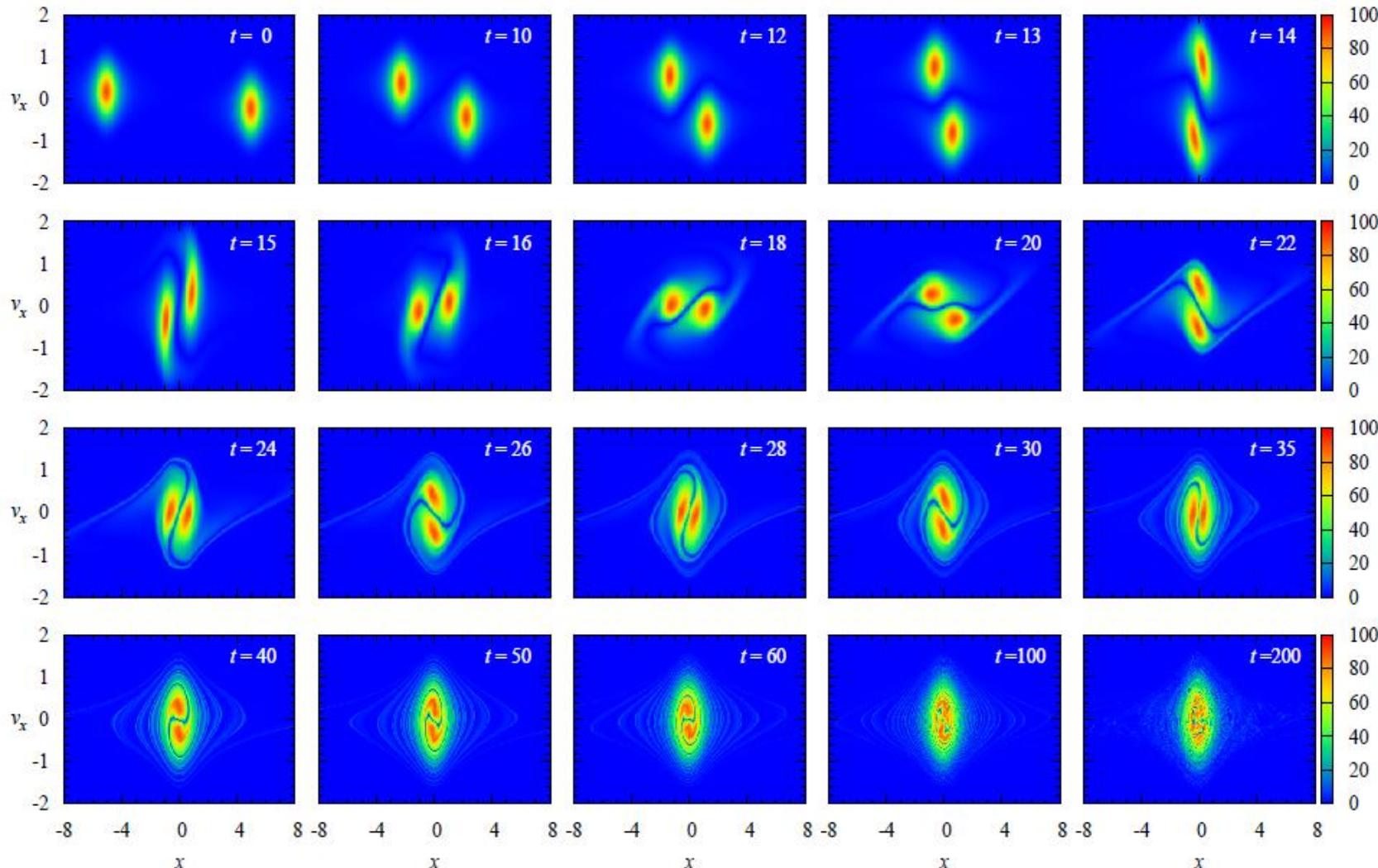
$f(r, u, j, t)$ at $j = 0.0015$



$f(r, u, j, t)$ at $j = 0.3$



Application of SCF Method: Phase-Space Evolution of Head-On Colliding Systems



$x-v_x$ plane with
 $y=z=0$ and $v_y=v_z=0$

● $N=10\,000\,584/\text{system}$

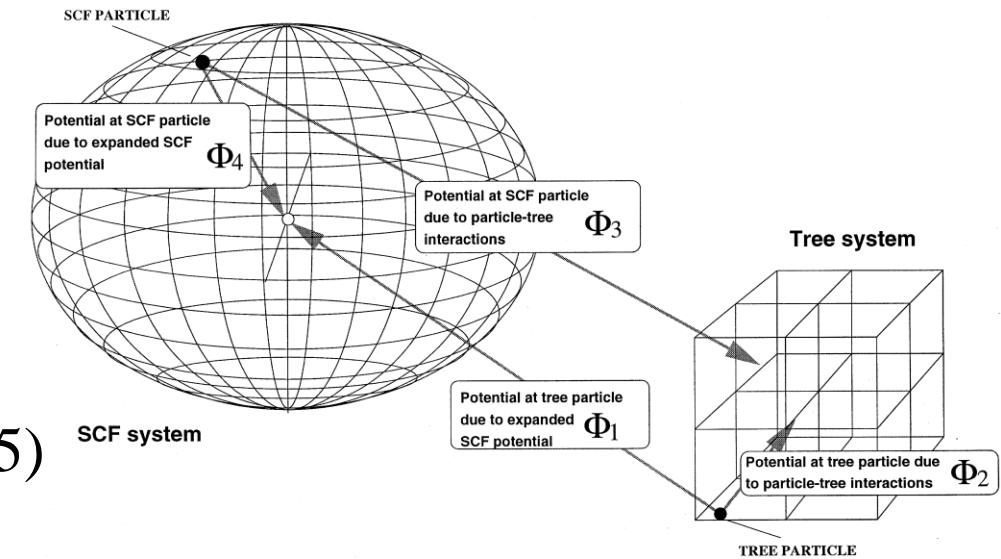
● self-gravity:
 $n_{\max}=16, l_{\max}=m_{\max}=10$

● interaction term:
 $n_{\max}=28, l_{\max}=m_{\max}=28$

SCF code と *tree code* の合体コード

- ・球状系では少ない展開項数で系を記述できる直交基底が存在
⇒ハロー粒子に **SCF法** を適用
- ・密度分布の局所構造の記述
⇒円盤粒子に **tree法** を適用
- ・*SCFTREE code*
 - Vine & Sigurdsson (1998, MNRAS, 295, 475)

重力作用元	重力作用先	手法
円盤粒子	円盤粒子	tree
ハロー粒子	ハロー粒子	SCF
ハロー粒子	円盤粒子	SCF
円盤粒子	ハロー粒子	tree

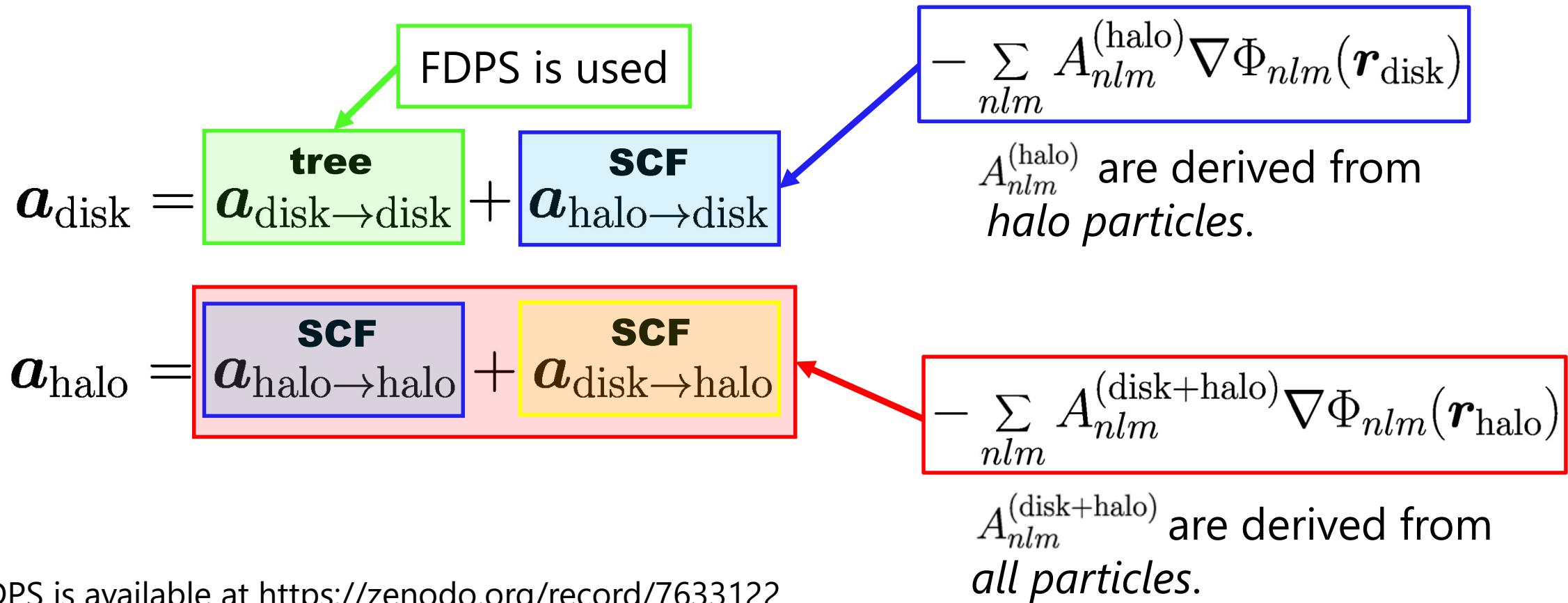


The potentials Φ_1 and Φ_4 are due to the expanded SCF potential, at a position relative to the center of mass of the SCF system. The potentials Φ_2 and Φ_3 are due to the tree system of particles.

SCF法で高速化: *SCF-FDPS*

SCF-FDPSコードの実現

- 加速度の計算



3. *SCF-FDPS* コードの性能評価

シミュレーションの詳細

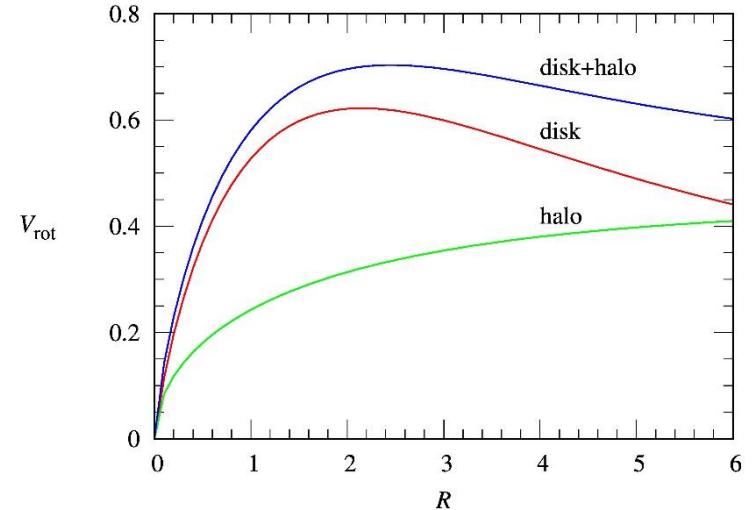
- Exponential disk

$$\rho_d(R, z) = \frac{M_d}{4\pi h^2 z_0} \exp(-R/h) \operatorname{sech}^2(z/z_0)$$

- NFW halo

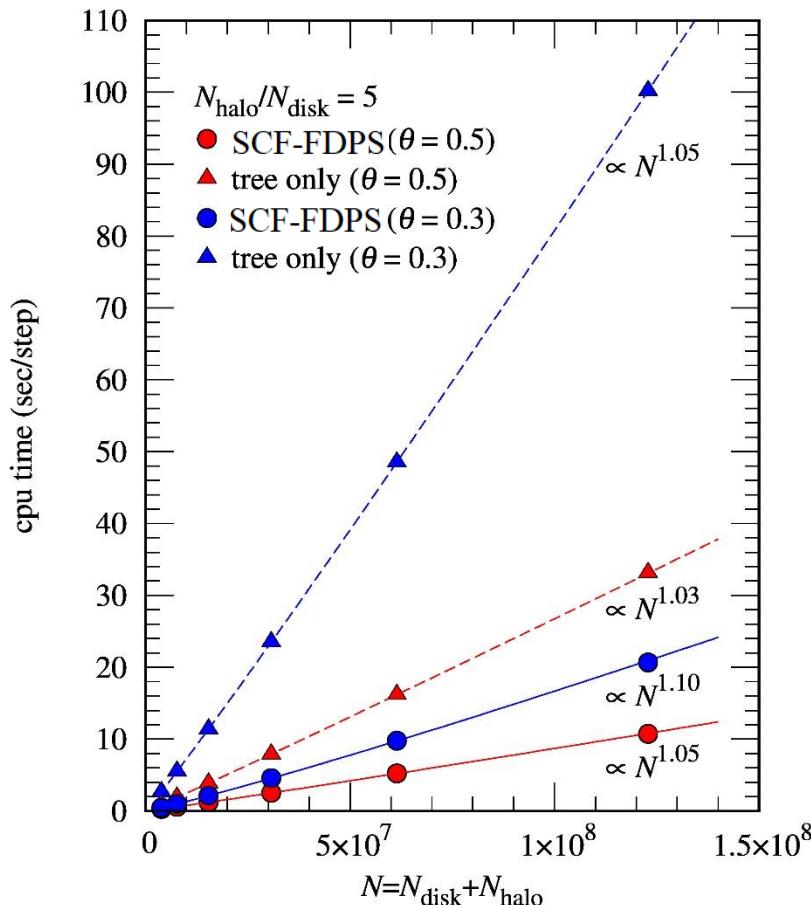
$$\rho_h(r) = \frac{\rho_0}{(r/r_s)(1+r/r_s)^2} \quad \left(\rho_0 = \frac{M_h}{4\pi R_h^3} \frac{C_{\text{NFW}}^3}{\ln(1+C_{\text{NFW}}) + C_{\text{NFW}}/(1+C_{\text{NFW}})}, C_{\text{NFW}} = R_h/r_s \right)$$

- Units: $G = M_d = h = 1$
- $R_h = 30, M_h = 5, C_{\text{NFW}} = R_h/r_s = 5$ ($r_s = 6$)
- Initial models are constructed using MAGI (*Miki & Umemura, 2018, MNRAS, 475, 2269*)
- tree: Plummer softening $\varepsilon = 0.006$
- tree: opening angle $\theta = 0.3, 0.5$
- tree: forces are expanded up to quadrupole order
- SCF: Hernquist-Ostriker's basis set with $n_{\max}=16, l_{\max}=16$, and $a_{\text{basis}}=6$
- time step: $\Delta t = 0.1$
- AMD Ryzen Threadripper 3990X 64-Core Processor

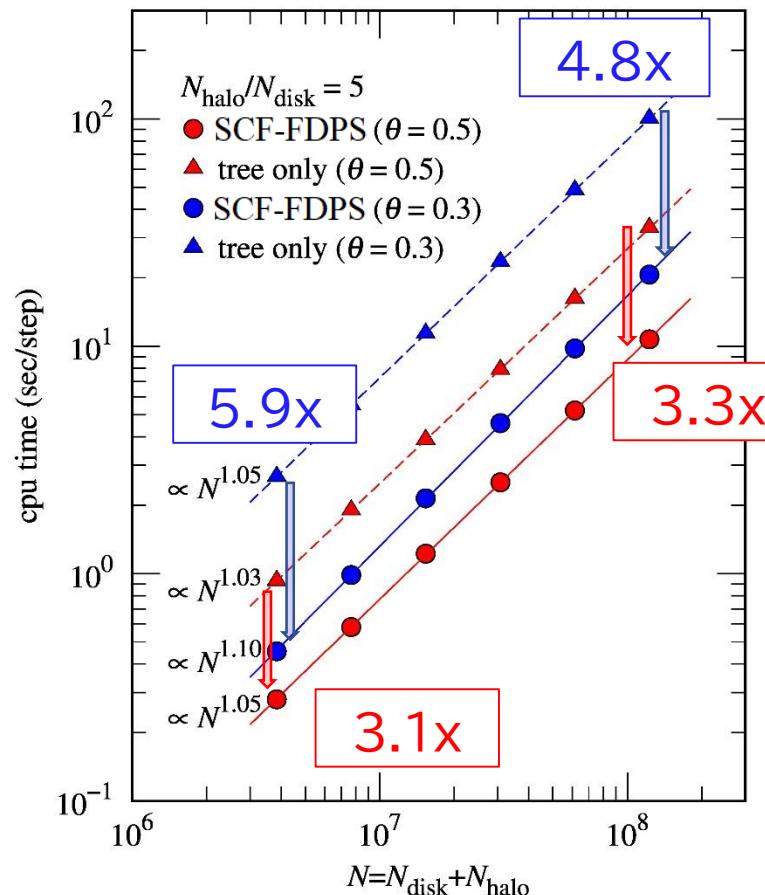


全粒子数と CPU time の関係

Linear Scale



Log Scale



AMD Ryzen Threadripper 3990X 64-Core Processor

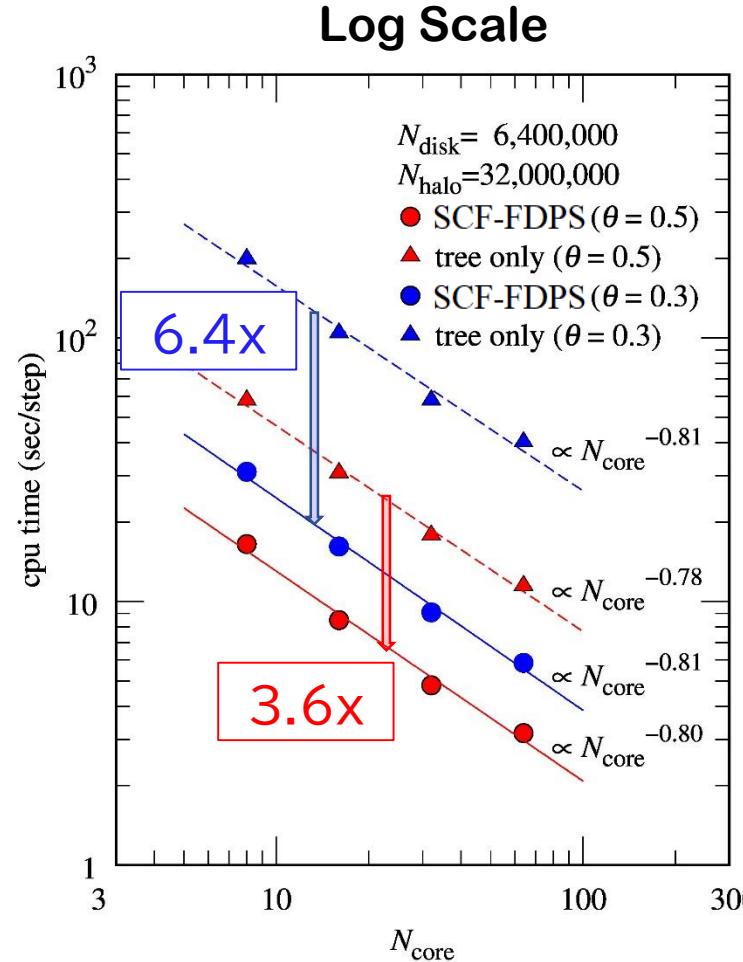
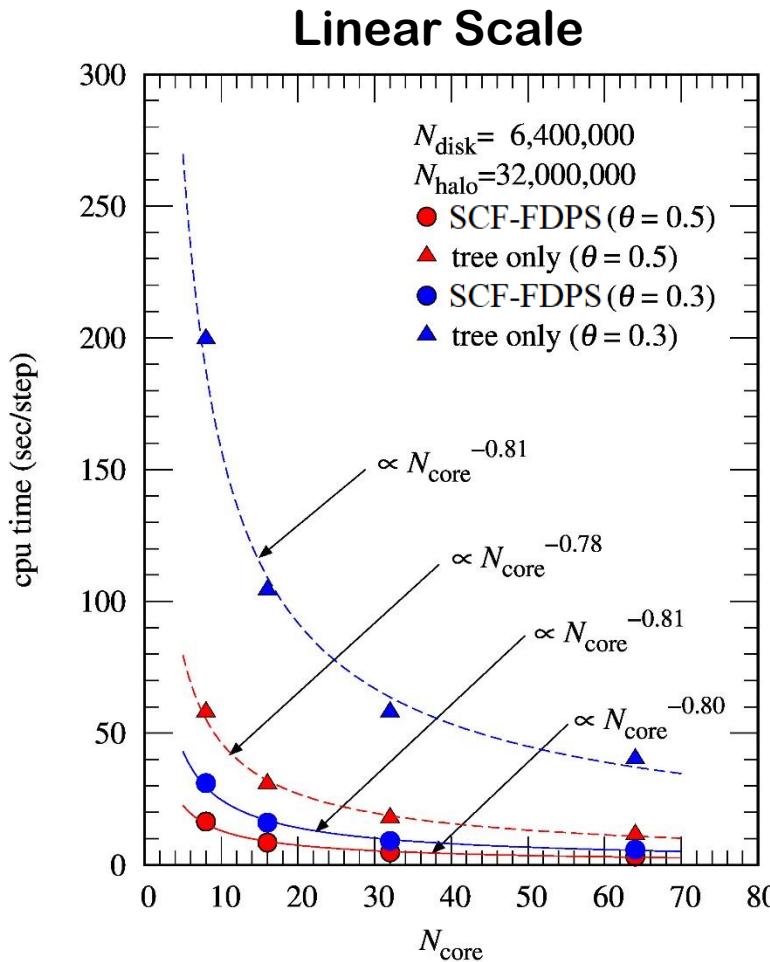
$N_{\text{halo}}/N_{\text{disk}} = 5$

N_{total}	N_{disk}	N_{halo}
3,840,000	640,000	3,200,000
7,680,000	1,280,000	6,400,000
15,360,000	2,560,000	12,800,000
30,720,000	5,120,000	25,600,000
61,440,000	10,240,000	51,200,000
122,880,000	20,480,000	102,400,000

$\theta=0.5$ で約3倍高速化

$\theta=0.3$ で約5~6倍高速化

コア数 N_{core} と CPU time の関係



$N_{\text{disk}} = 6,400,000$

$N_{\text{halo}} = 32,000,000$

$N_{\text{core}} = 8, 16, 32, 64$

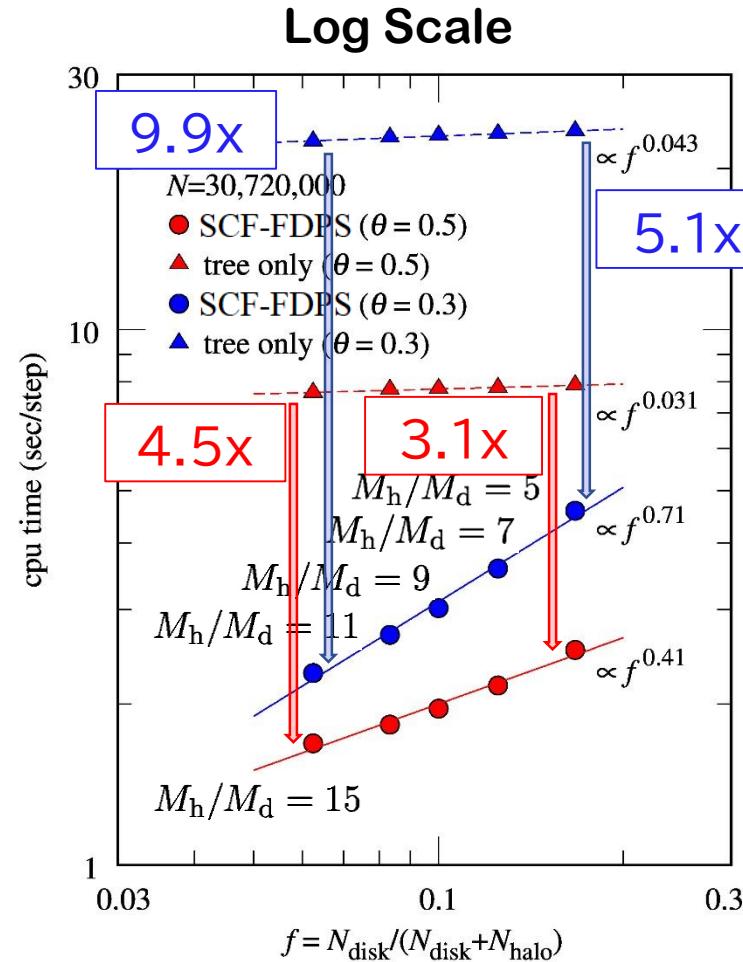
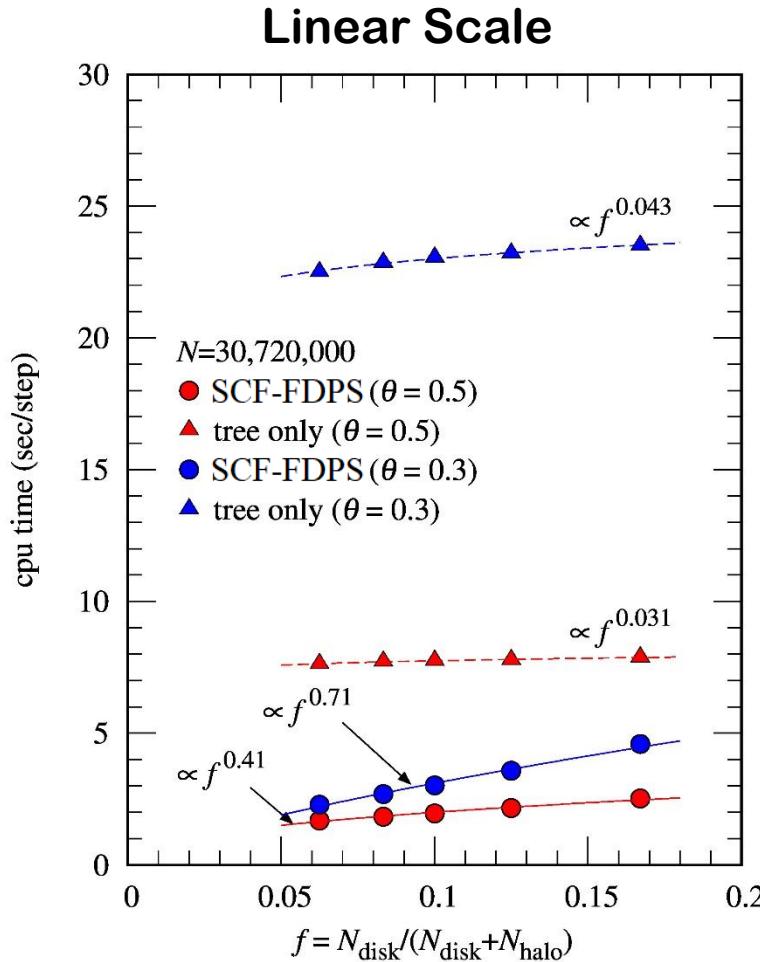
$\theta = 0.5$ で約3.6倍高速化

$\theta = 0.3$ で約6.4倍高速化

※CPU clock is lowered as N_{core} increases

AMD Ryzen Threadripper 3990X 64-Core Processor

ツリー粒子の割合と CPU time の関係



$$N_{\text{total}} = 30,720,000$$

$$M_{\text{halo}}/M_{\text{disk}} = 5, 7, 9, 11, 15$$

$$R_h = 30, C_{\text{NFW}} = 5$$

$$f = N_{\text{disk}}/N_{\text{total}}$$

f	N_{disk}	N_{halo}
0.0625	1,920,000	28,800,000
0.0833	2,560,000	28,160,000
0.1	3,072,000	27,648,000
0.125	3,840,000	26,880,000
0.167	5,120,000	25,600,000

$\theta=0.5$ で約3~4.5倍高速化

$\theta=0.3$ で約5~10倍高速化

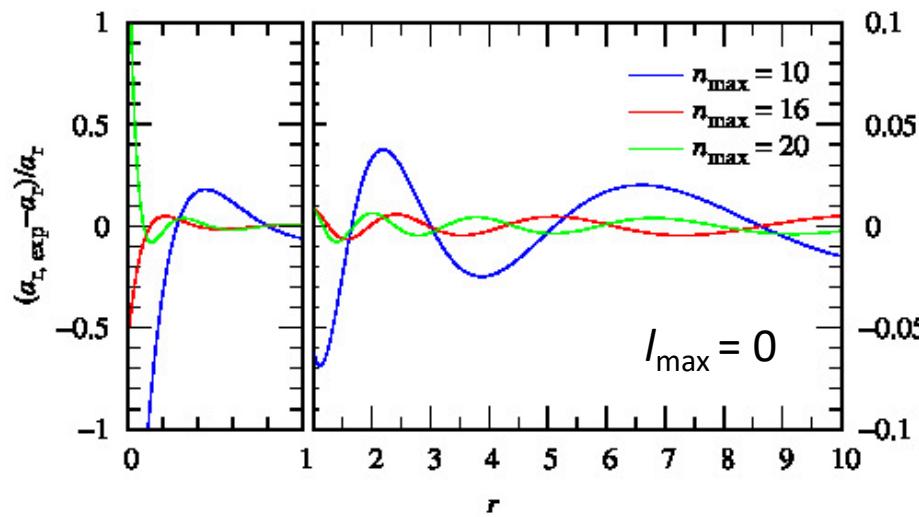
AMD Ryzen Threadripper 3990X 64-Core Processor

4. シミュレーション結果の評価

ハローの展開項数 n_{\max} , l_{\max} の指標

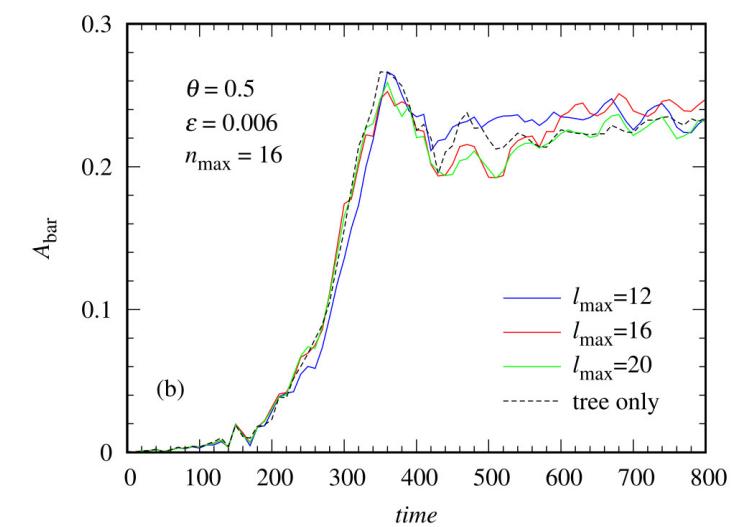
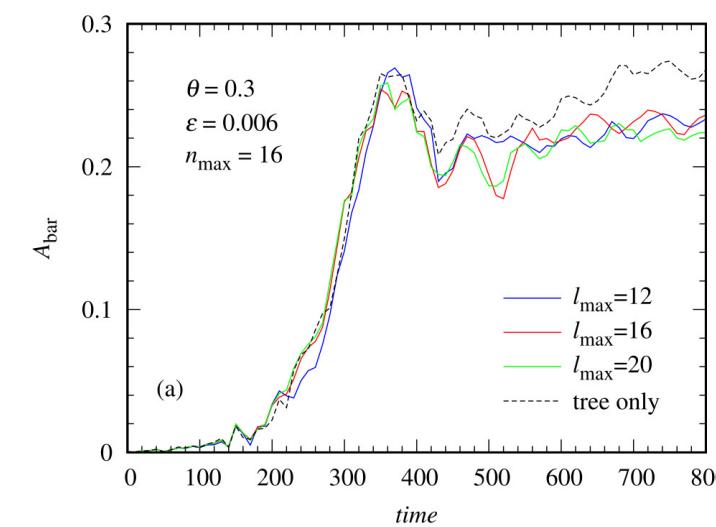
- n_{\max} の指標

球対称モデルの半径方向の加速度



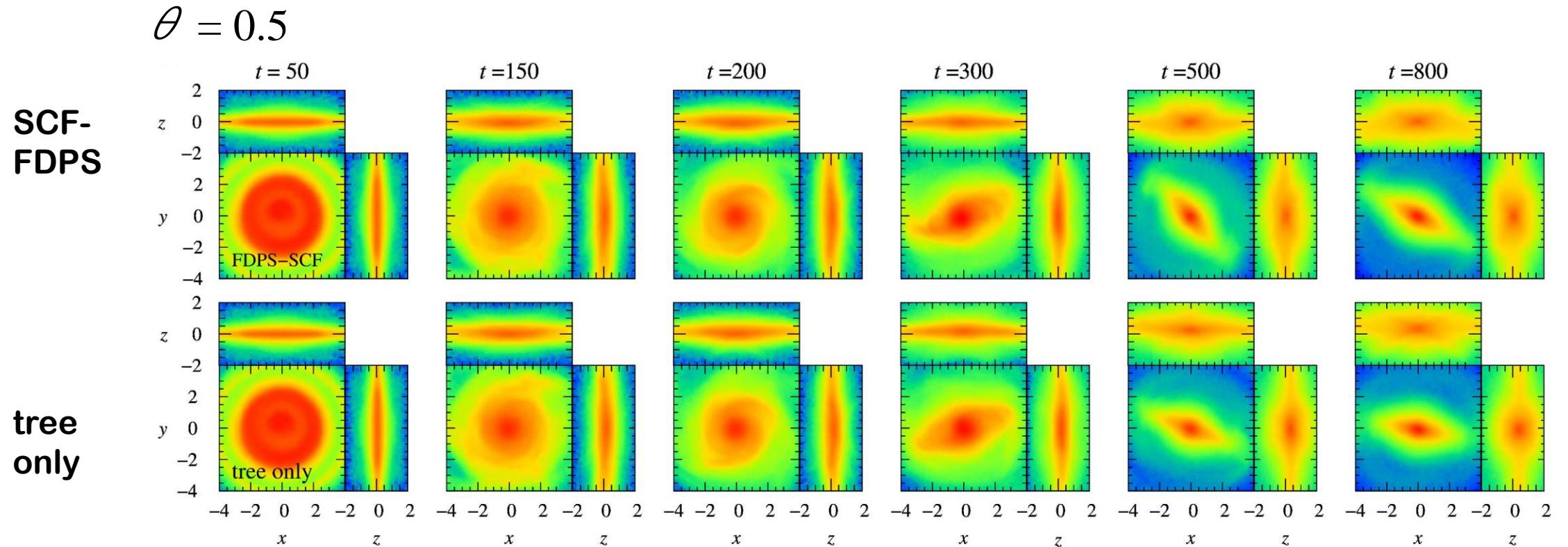
- l_{\max} の指標

バー振幅の時間進化



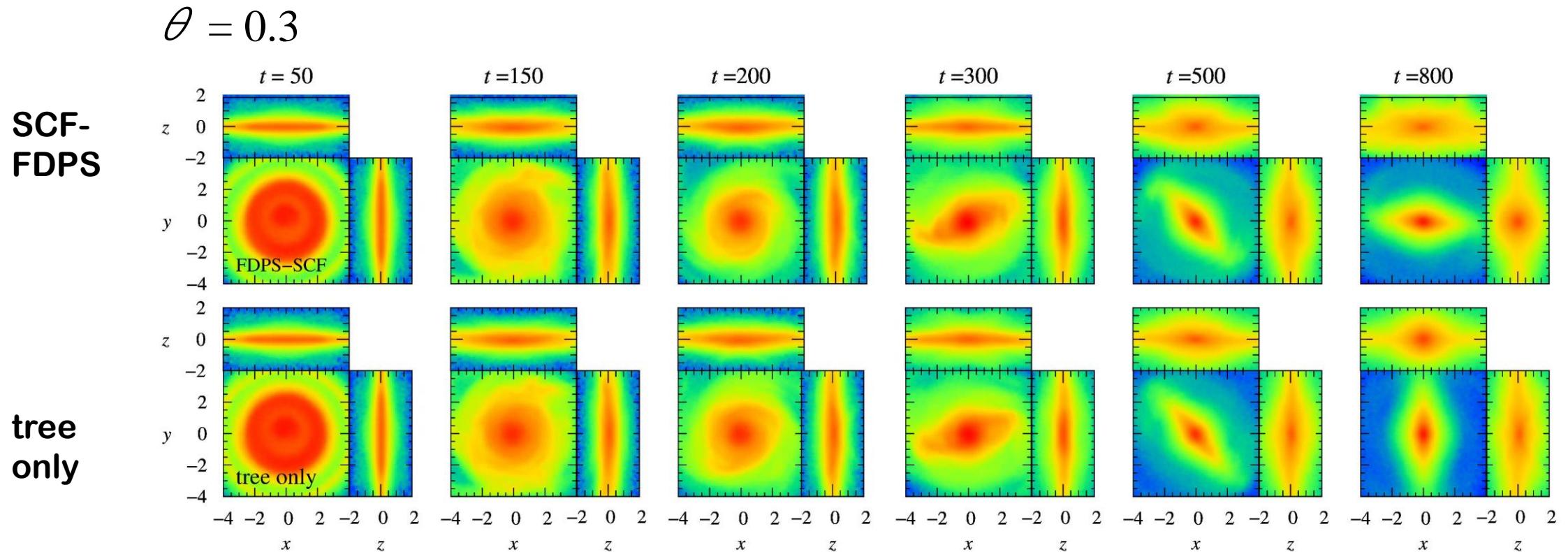
$N_{\text{disk}}=6,400,000$, $N_{\text{halo}}=32,000,000$

Time evolution of projected densities



$N_{\text{disk}}=6,400,000, N_{\text{halo}}=32,000,000, \theta=0.5, \varepsilon=0.006$

Time evolution of projected densities



$N_{\text{disk}}=6,400,000, N_{\text{halo}}=32,000,000, \theta=0.3, \varepsilon=0.006$

5. まとめ

- ・円盤とハローからなる系のシミュレーションのための高速な N 体計算コードである*SCF-FDPS*コードを開発した
 - ・円盤の自己重力のみtree法で計算
 - ・ハローの自己重力とハロー粒子と円盤粒子の相互作用をSCF法で計算
- ・*FDPS*によるツリーコード単独の計算より3倍から10倍程度の高速化を実現した
- ・計算時間はほぼ全粒子数に比例する
- ・計算時間はほぼコア数に反比例数する
- ・円盤粒子数の割合 $f=N_{\text{disk}}/N_{\text{total}}$ が増えれば計算時間は長くなるが、*FDPS*によるツリーコード単独で計算する場合に比べて少なくとも約4倍以上高速
- ・円盤の密度の時間進化は*FDPS*によるツリーコード単独で計算したものと同等
- ・*Plummer softening* と *spline softening* を実装済み
- ・*Hernquist & Ostriker's basis set* (密度分布:cuspy)と *Clutton-Brock's basis set*(密度分布:cored) を実装済み