Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

PINOCCHIO Code: Latest Developments and GPU Transition

Marius D. Lepinzan, P. Monaco, T. Castro and L. Tornatore

Spoke 3 II Technical Workshop, Bologna Dec 17 -19, 2024

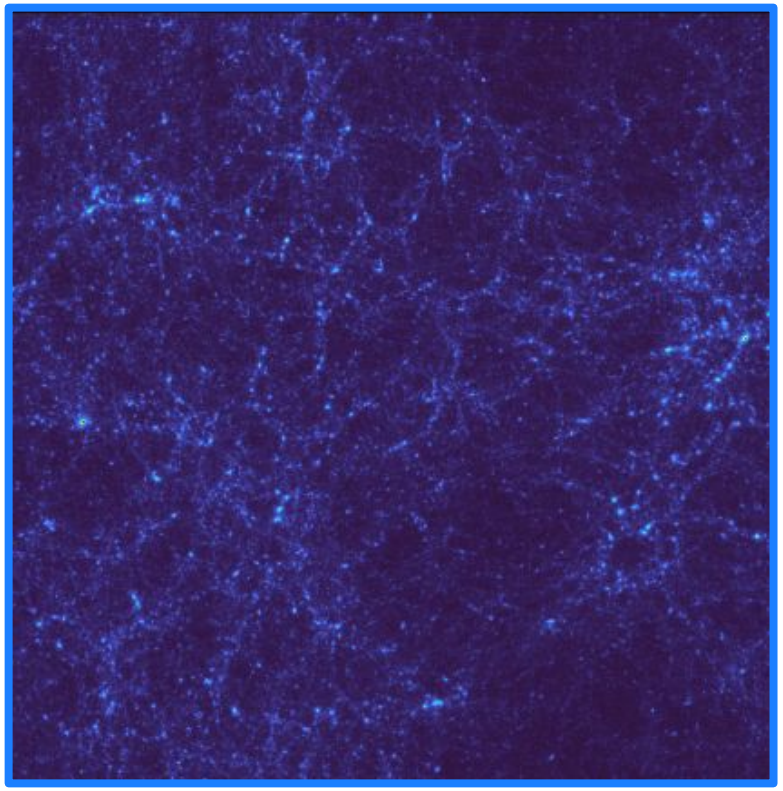
Scientific Rationale

PINOCCHIO is a code, based on Lagrangian Perturbation Theory (LPT), for simulating Dark Matter halos in cosmological boxes and past light cones (*Monaco et al. 2002, 2013; Munari et al. 2017)*

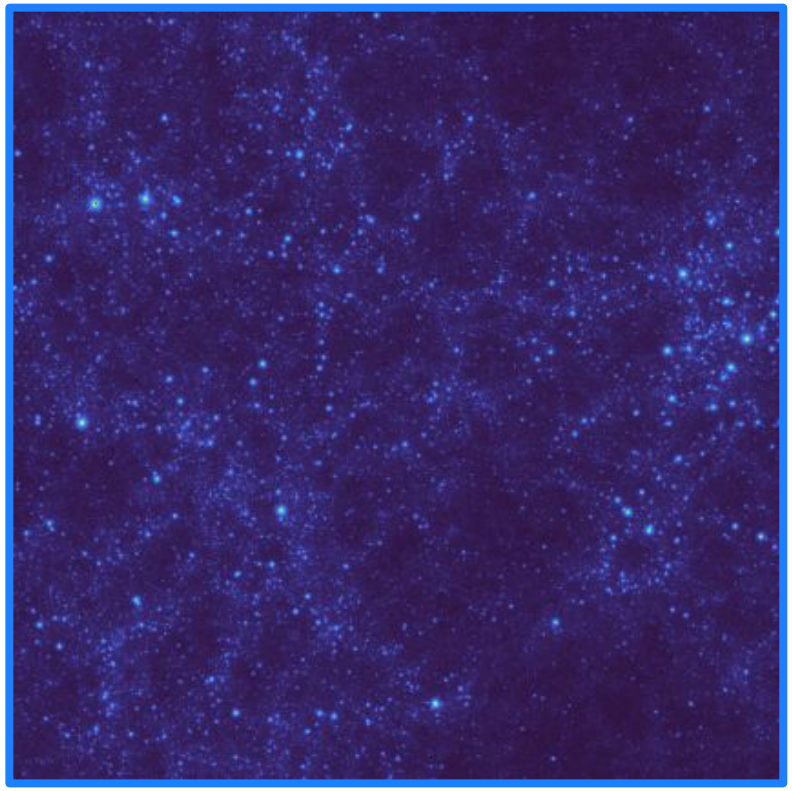
Comparison with full N-body simulations:

- ~1000 faster
- 5 – 10% accuracy in reproducing 2-point halo statistics, halo mass function and halo bias
- 5 – 10% accuracy in reproducing cosmic void statistics (*Lepinzan et al. in prep)*

GADGET



PINOCCHIO



Technical Objectives, Methodologies and Solutions

- **Optimize the code to fully leverage modern HPC infrastructure, including GPUs:**

  - **Improve code performance**: *suitable threadization? main bottlenecks ?*

  - **Identify off-loadable regions**: *what can be ported to GPUs?*

  - **Improve scientific output**: *Adopting new algorithm?*

- **Adopted solutions:**

  - **Improve** the **MPI** framework: **OpenMP**

  - **Porting** collapse times to **GPU**: **OpenMP**

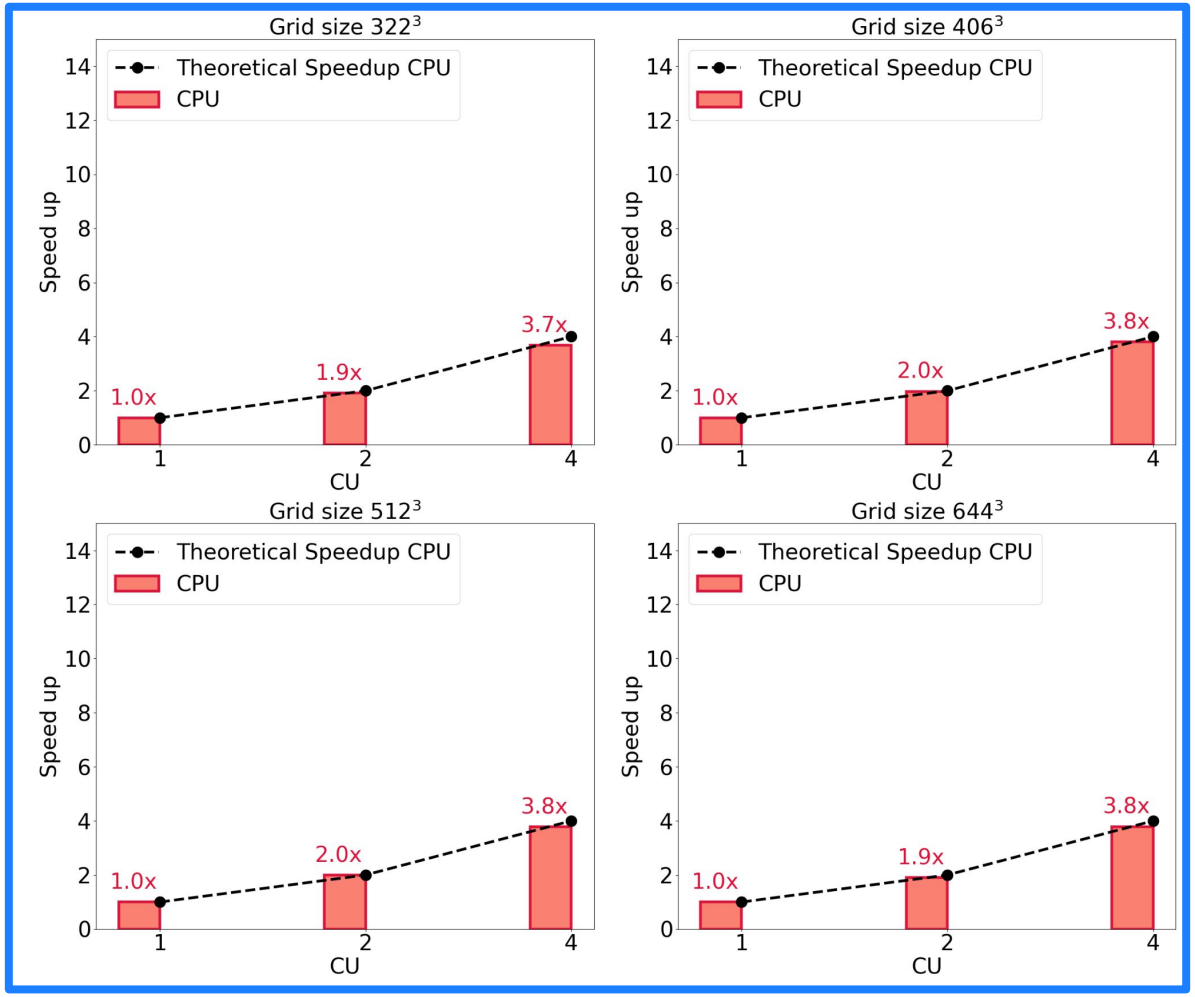
  - **Optimize** and **investigate** a new fragmentation algorithm: **ADP vs HDBSCAN**

  - **Testing, bug fixing, testing, bug fixing… !!**

Main Results

Extending the existing parallel computing paradigm by integrating OpenMP into the collapse times calculation

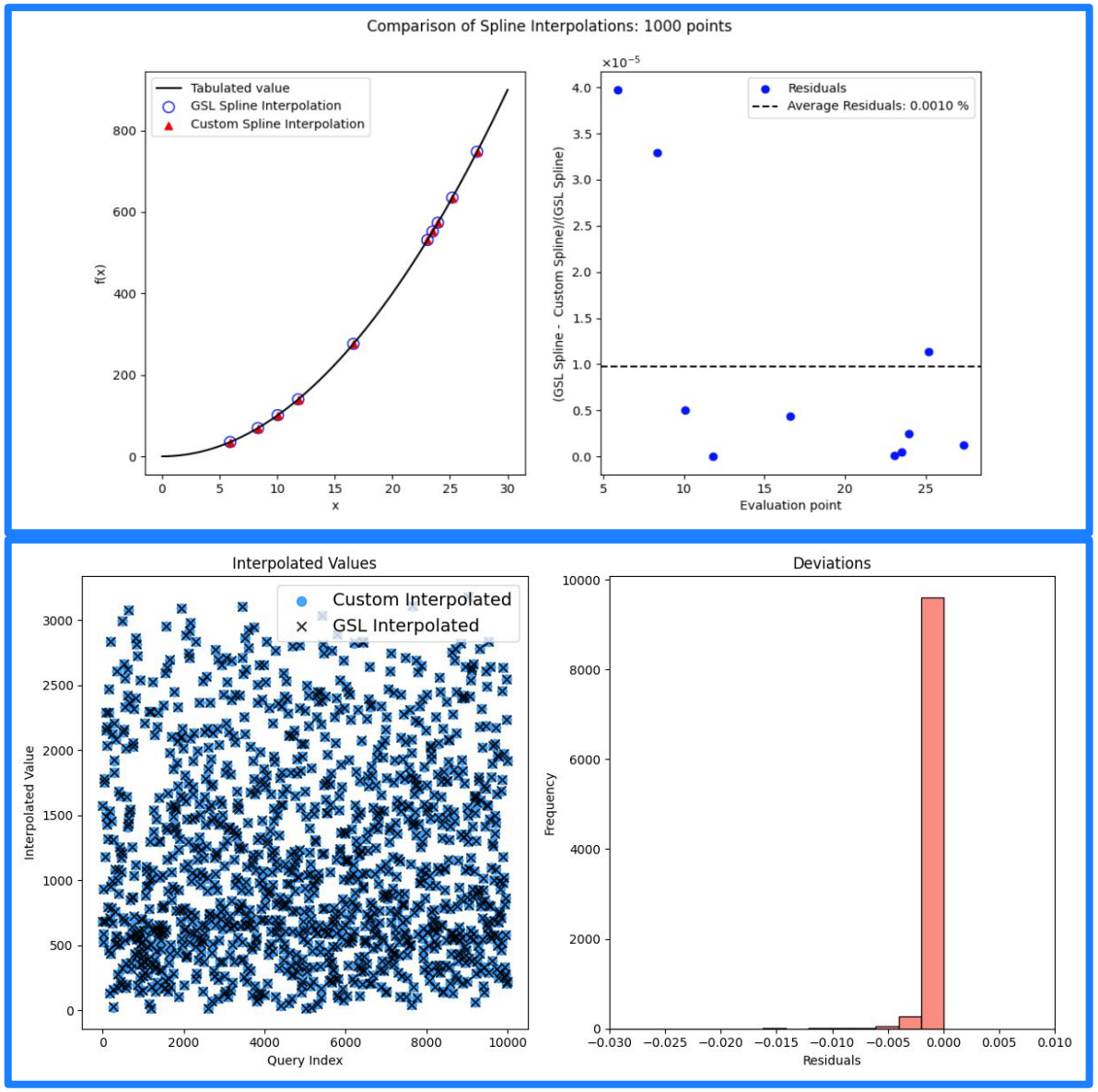
- Nearly **ideal** scaling up
- Large Euclid Box (box ~ 4 Gpc, 4096^3 particles) **computational time: ~ 8%** out of **~ 40 minutes**
- Computational time **improvement: ~ 4x speed-up**
- Thousands of mocks: **~ 40 human hours less**



Main Results

Offloading of collapse times calculation on GPU with OpenMP:

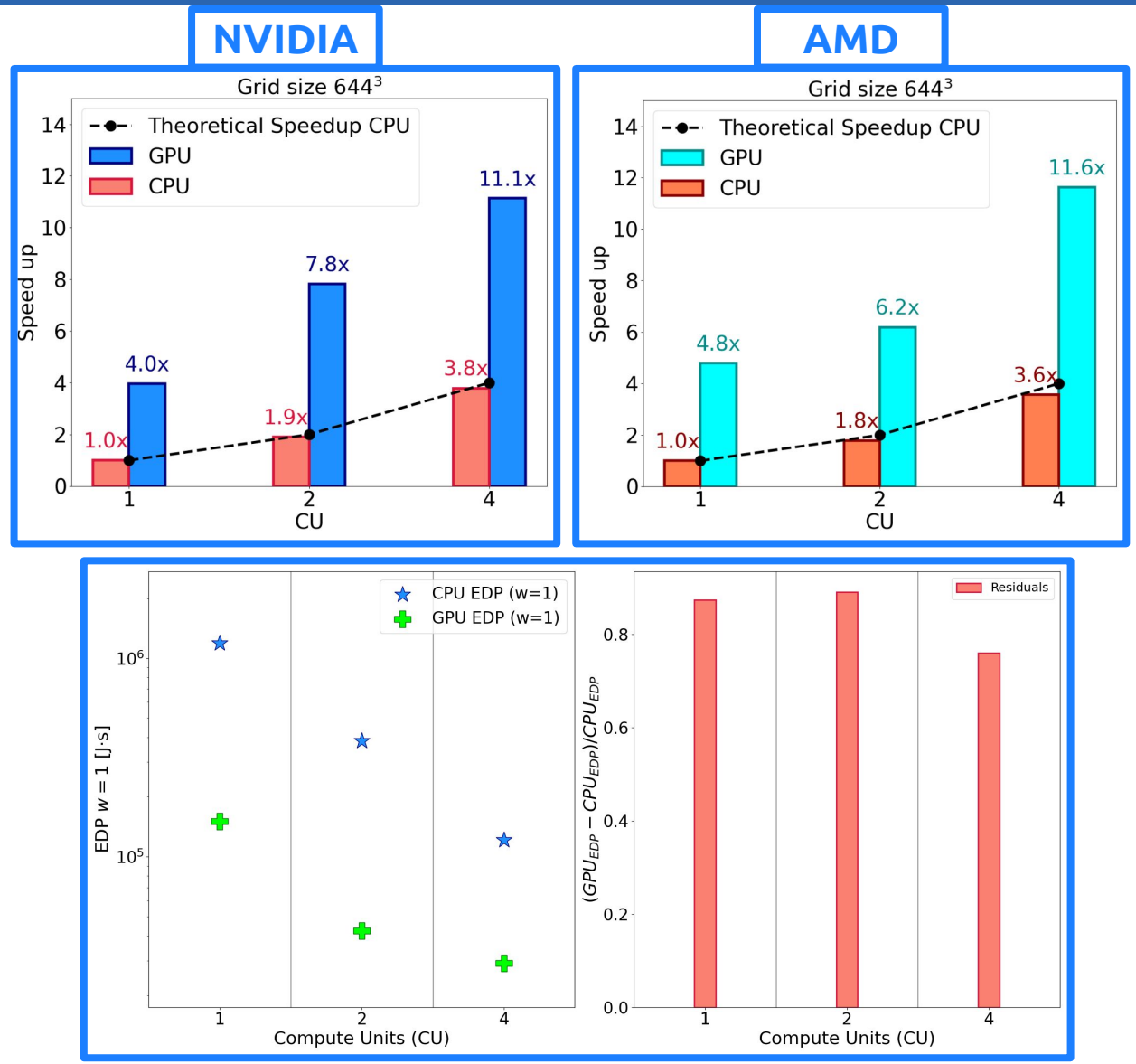
- **Offloading main issue:** need of a **custom cubic spline** and **bilinear spline** interpolation
- GPU **offloading** test out of PINOCCHIO and comparison with GSL: **done**
- **Integration** in PINOCCHIO and **test** of **GPU vs CPU** final **scientific output**: minor differences that **do not impact** the code **primary** usage



Main Results

Offloading of collapse times calculation on GPU with OpenMP:

- **GPU cubic spline** version tested on **NVIDIA** and **AMD** platforms: **offloading** and **performance portability achieved** (***~10x speedup***)
- **Power consumption measurements** integrated for both **CPUs** and **GPUs**: Power Measurement Toolkit (**PMT**) only on **NVIDIA** platform (**GPU** kernel ***~80 %*** more **efficient**)
- **GPU bilinear spline** still to be optimized: main issue **memory transfer**
- **PDP proceeding** ***(Lepinzan et al. sub)*** and technical **paper** (***Lepinzan et al. in prep)***

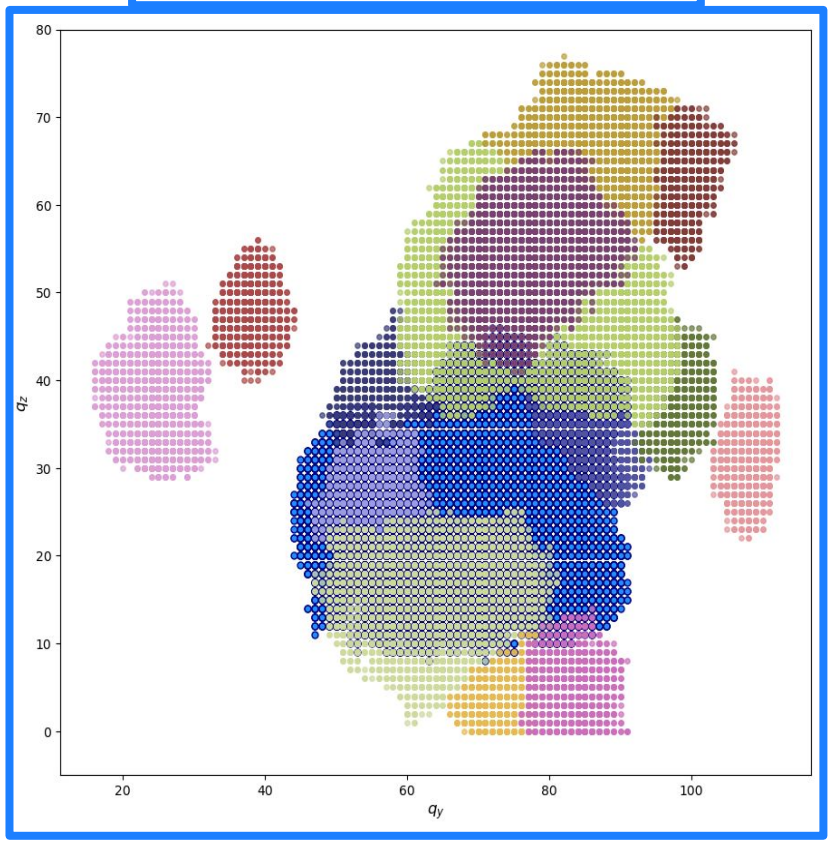


Main Results

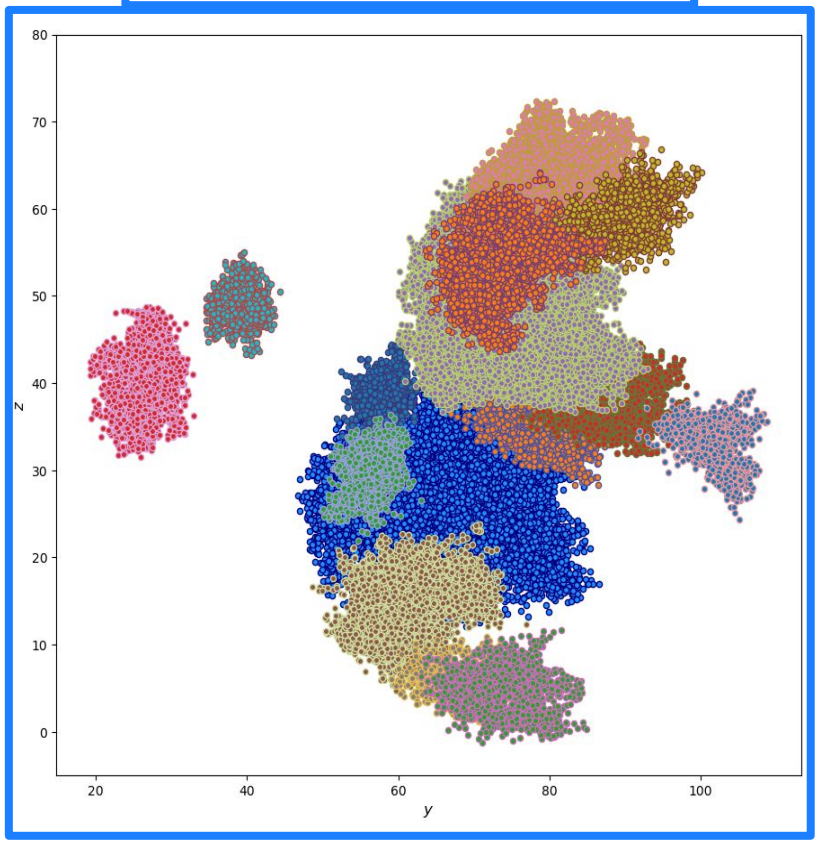
New methodology for the fragmentation (halo reconstruction)

- **Clustering algorithm** (Advance Density Peak) for a **domain decomposition**: identify **Eulerian patches** that will **end up in halos** according to **PINOCCHIO**
- Apply the **current algorithm** for **fragmentation** on every **independent domain**

Lagrangian patches

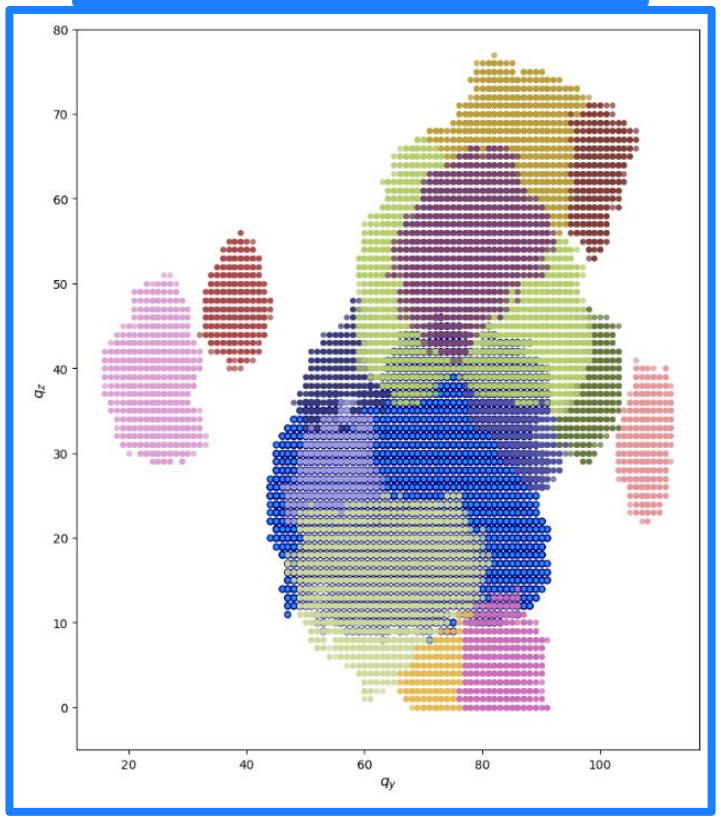


Eulerian patches

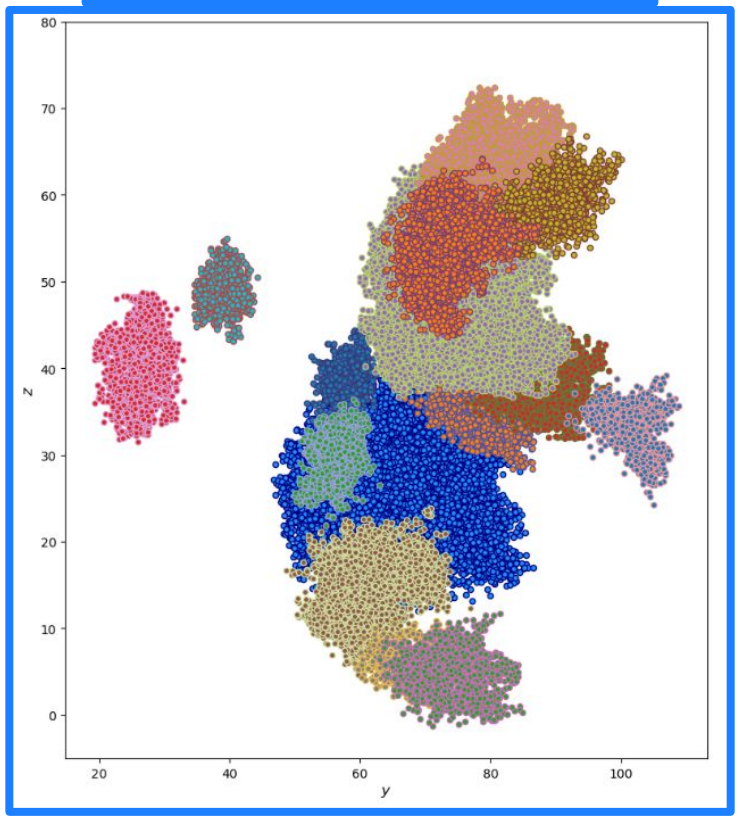


Main Results

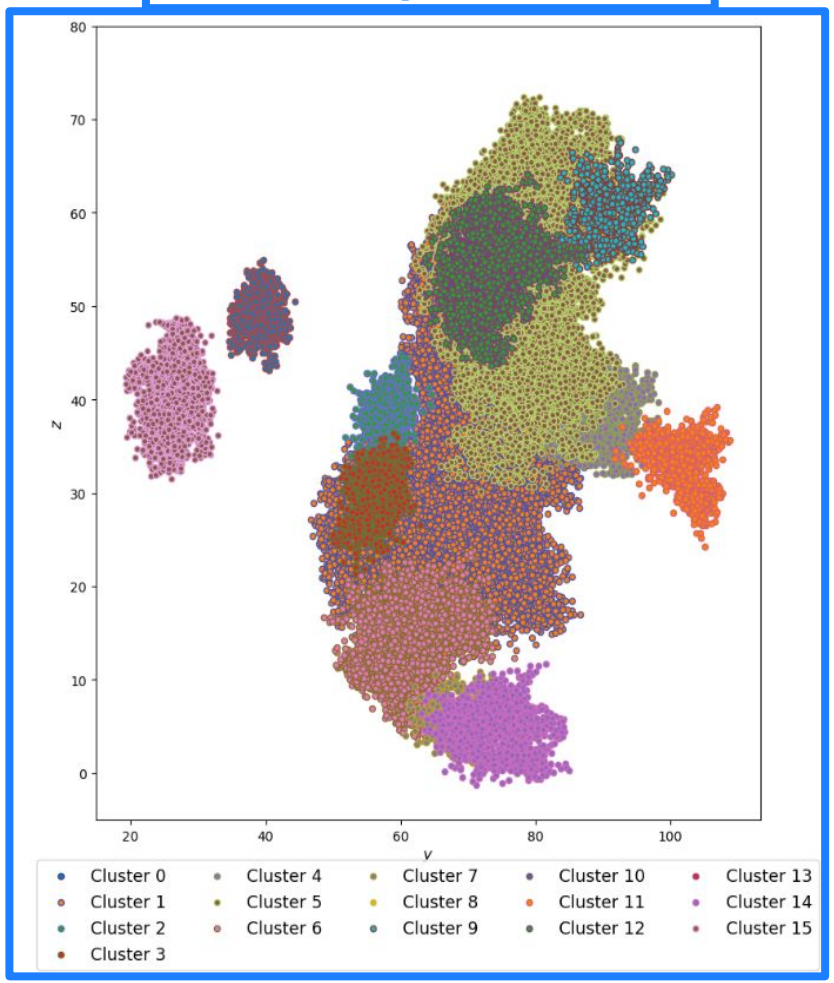
Lagrangian patches



Eulerian patches



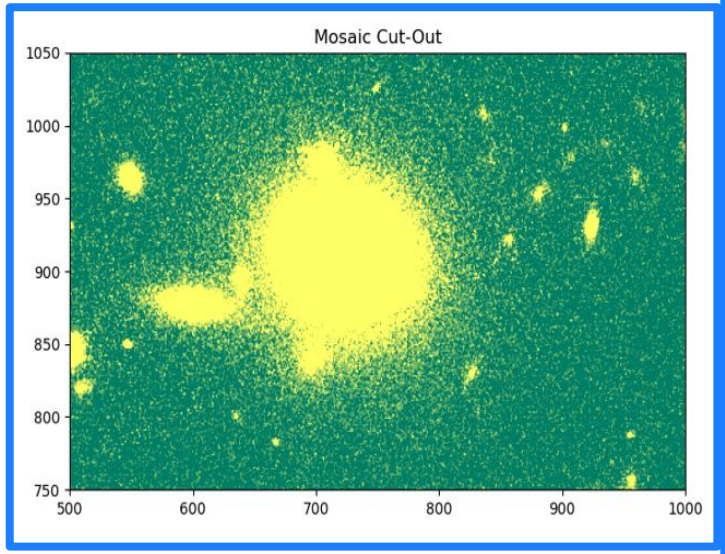
ADP patches

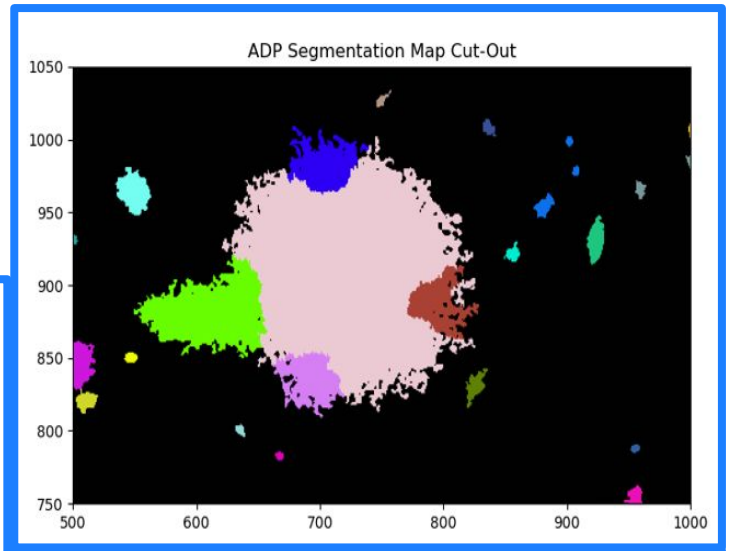


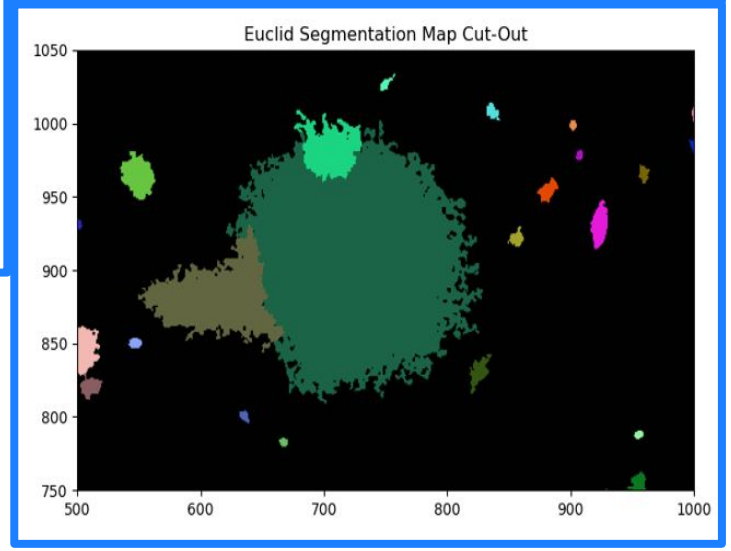
Main Results

New methodology for the fragmentation (halo reconstruction)

- **Bypass** the **Eulerian space** and apply the **clustering algorithm** directly to a regular **3D grid** of points, using the **collapse time** for each particle provided by PINOCCHIO



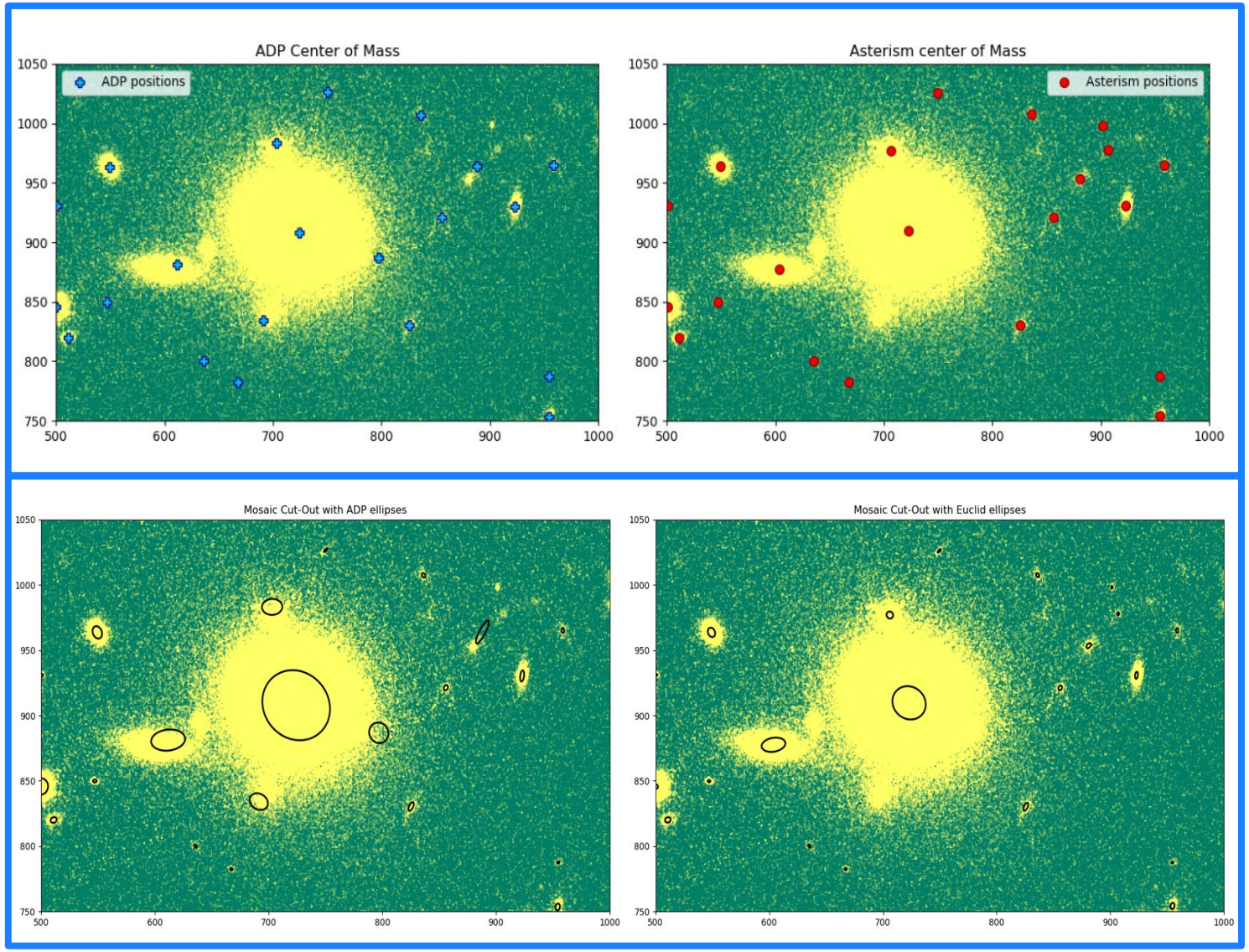




Main Results

New methodology for the fragmentation (halo reconstruction)

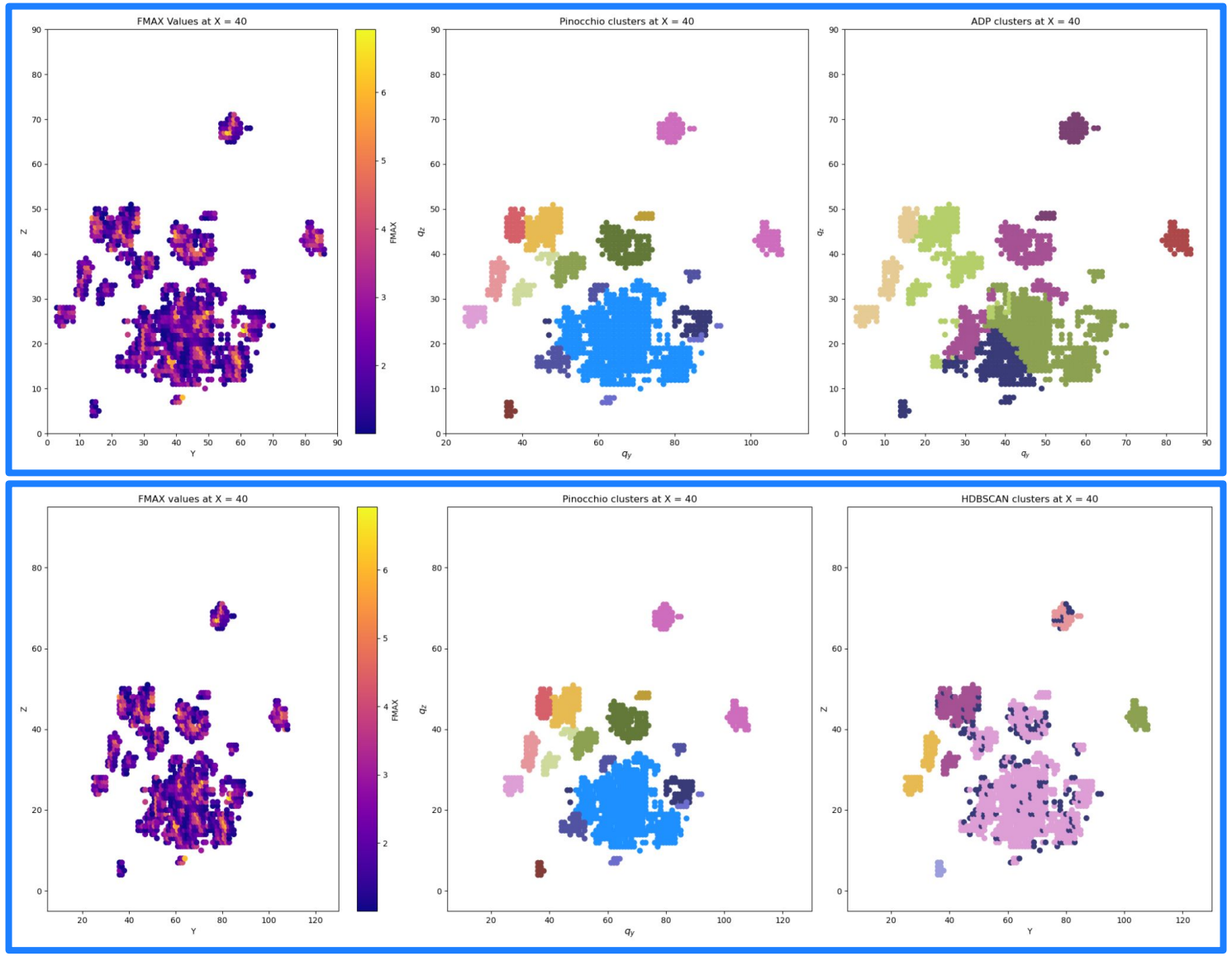
- **2D version of ADP already implemented and tested for source Debleding on a simulated image of True Universe (TU) against official *Euclid* algorithm**



Main Results

New methodology for the fragmentation (halo reconstruction)

- **Extension** to a **3D** version of **ADP implemented**
- **Only** collapse times information (**FMAX**)
- **Comparison** with other **clustering algorithm**: **HDBSCAN**



Timescale, Milestones and KPIs

04/2023 | 09/2023 | 02/2024 | 06/2024 | 10/2024 | 08/2025

MS 6

1. Familiarization with the code
2. Bug fixing
3. Identification off-loadable regions and bottleneck

MS 7

1. Threadization of collapse time with OpenMP
2. GPU offloading of collapse times with OpenMP: delayed because of GSL incompatibility

MS 8

1. GPU offloading of collapse times with OpenMP: done
2. Machine learning approach for fragmentation: ongoing
3. New stable release of the code (v5.1): normal, neutrinos and modified gravity (MG) test done

MS 9

1. Collapse times GPU: Multi-MPI, Multi-GPU version tested on NVIDIA and AMD platforms
2. Machine learning approach: ADP vs HDBSCAN on going
3. Stable 5.1 version: ready without MG

MS 10

1. New fragmentation
2. FFTs on GPUs
3. 100 simulations up 1000^3 particles
4. Technical paper on the code

Final Steps

- Add velocities information to the 3D grid-based version of ADP
- Test the 3D grid-based version of HDBSCAN with velocities information
- Optimize of the GPU version of tabulated collapse (custom bilinear spline) times calculation by adopting a full GPU interpolation procedure
- New code documentation
- Euclid-like simulations with the GPU version of collapse times calculation and new fragmentation