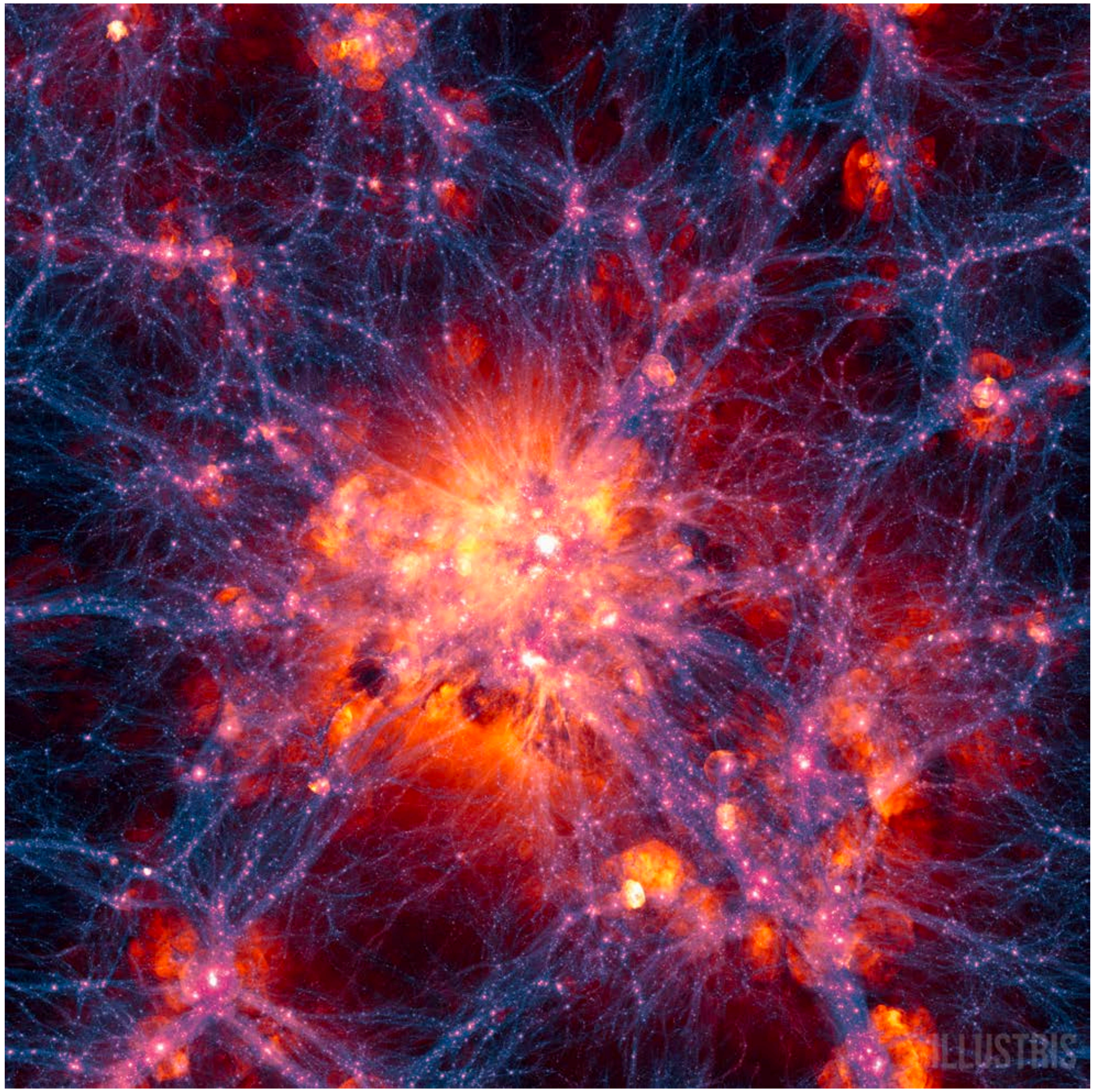
Large Scale Simulations



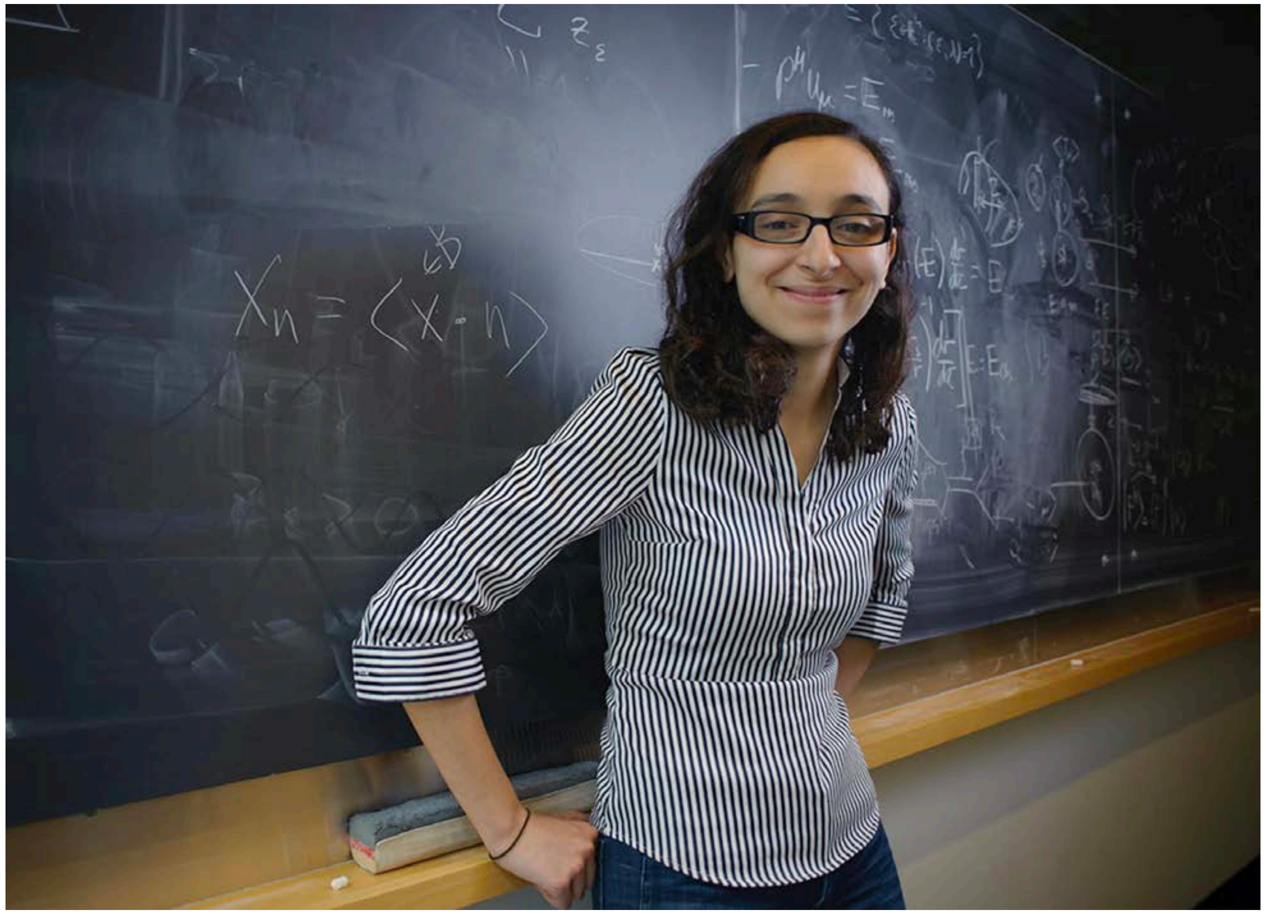
Our Goal In Lecture

- Going to build up the intuition for large-scale simulation
  - Focus on the simulation of galaxies to the universe
  - This lecture will give the tools to do this
- The devil is in the details
  - Our lecture use the core concepts in large scale sim
  - To get everything to work at scale is much harder
- We will touch on where the field is going

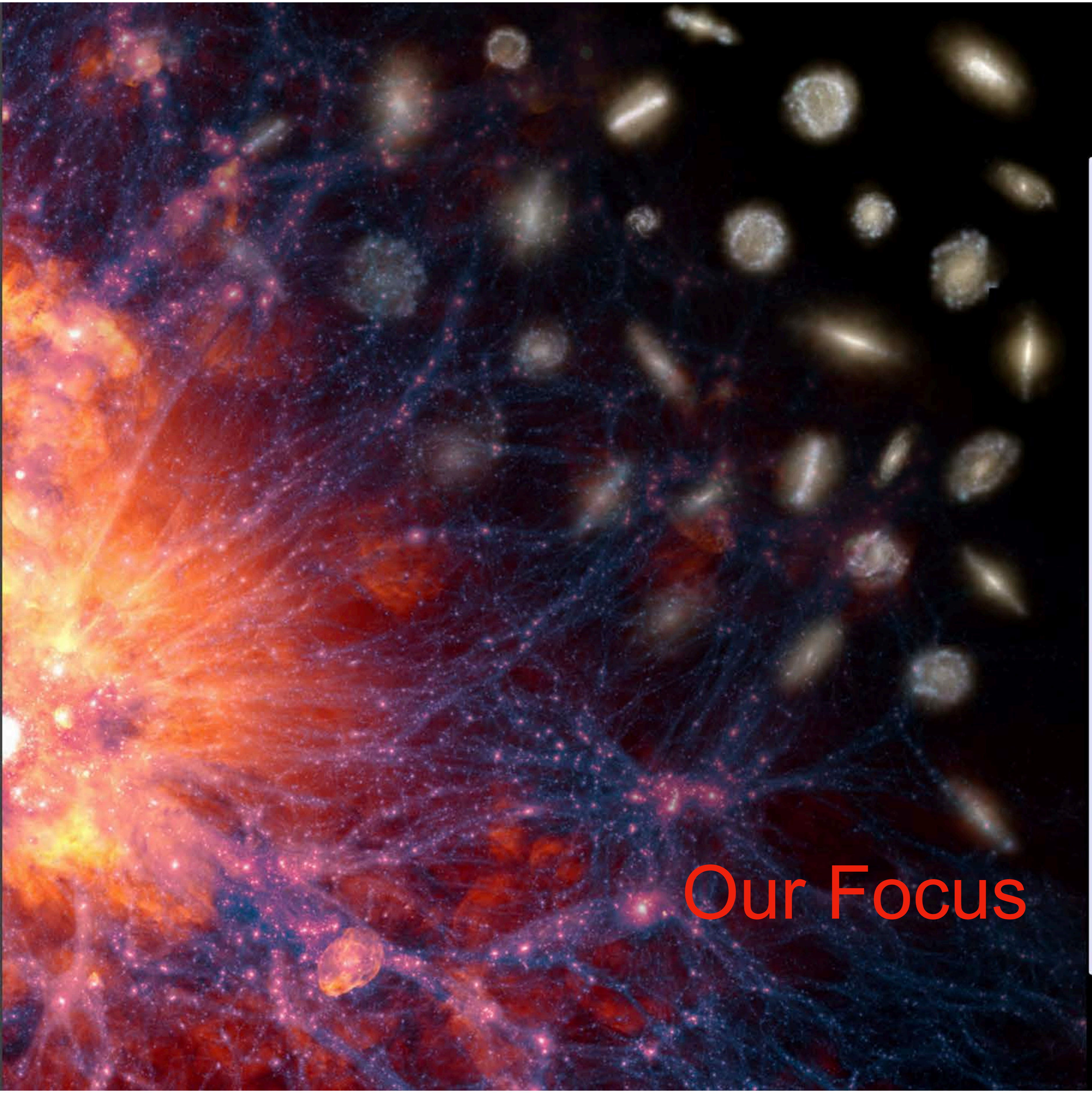
Lets Take a look at the Scale of Things

- https://www.illustris-project.org/media/

Prof. Mark Vogelsberg
Main Inspiration for talk



Prof. Lina Necib
Doing related work on smaller scales



Illustris/IllustrisTNG Model:
- basic ingredients -

- **hydrodynamics:**
  quasi-Lagrangian moving mesh (Arepo, Springel 2010)
- **heating / cooling:**
  primordial, metal line
- **UV background:**
  with self-shielding correction
- **star formation / ISM:**
  effective EOS
- **chemical enrichment:**
  9 elements by SNIa, SNII, AGB
- **supernova feedback:**
  kinetic SNII feedback
- **supermassive black holes:**
  seeding, growth, merging
- **AGN feedback:**
  quasar, radio mode, radiative

MV+ 2013, 2014

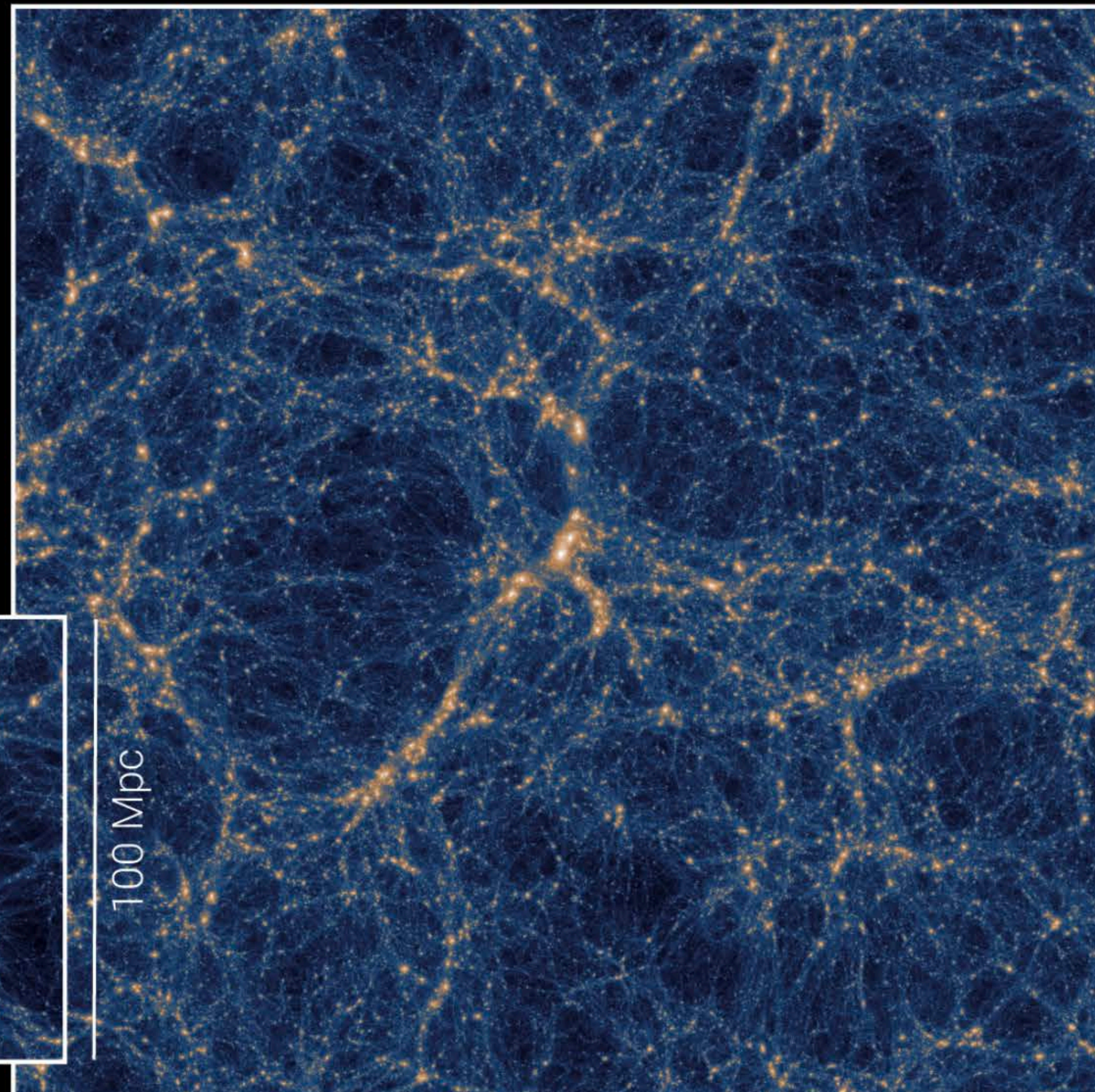
IllustrisTNG Team:

Mark Vogelsberger
Shy Genel
Volker Springel
Paul Torrey
Lars Hernquist
Dylan Nelson
Rainer Weinberger
Federico Marinacci
Ruediger Pakmor
Annalisa Pillepich
Jill Naiman

Illustris Team

three boxes with different primary science focus
(~250 million CPUh)

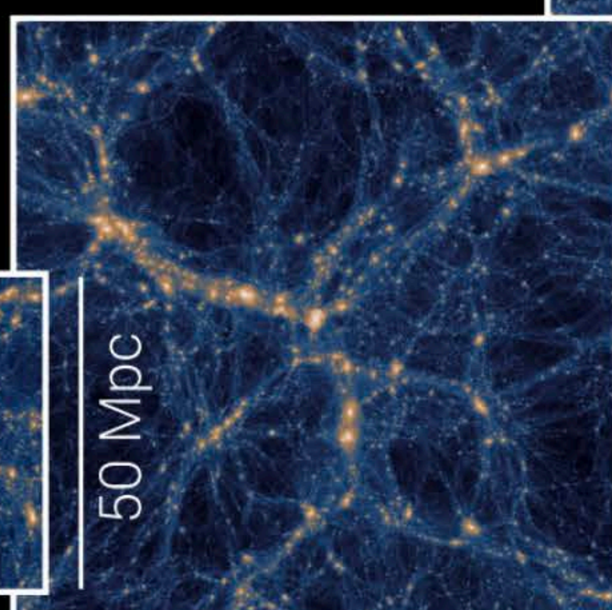
TNG300



300 Mpc

galaxy clusters

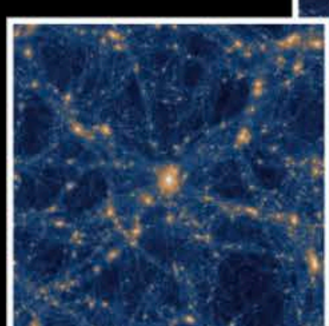
TNG100



100 Mpc

L* galaxies

TNG50



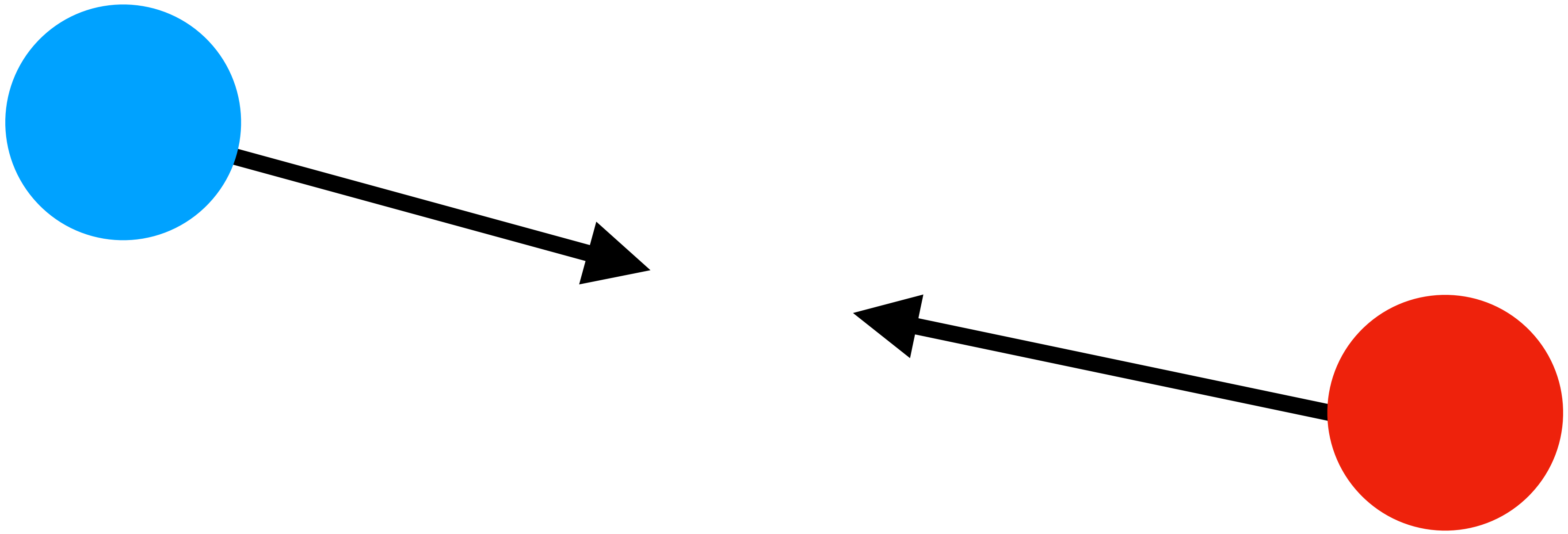
50 Mpc

dwarfs

Details

- This lecture is built on the following references:
    - https://www.youtube.com/watch?v=Fo23ihGLPA0
    - https://td.lpi.ru/~eugvas/nbody/lectures.pdf
    - https://www.tat.physik.uni-tuebingen.de/~schaefer/teach/f/chaos_english.pdf
    - https://blbadger.github.io/3-body-problem.html

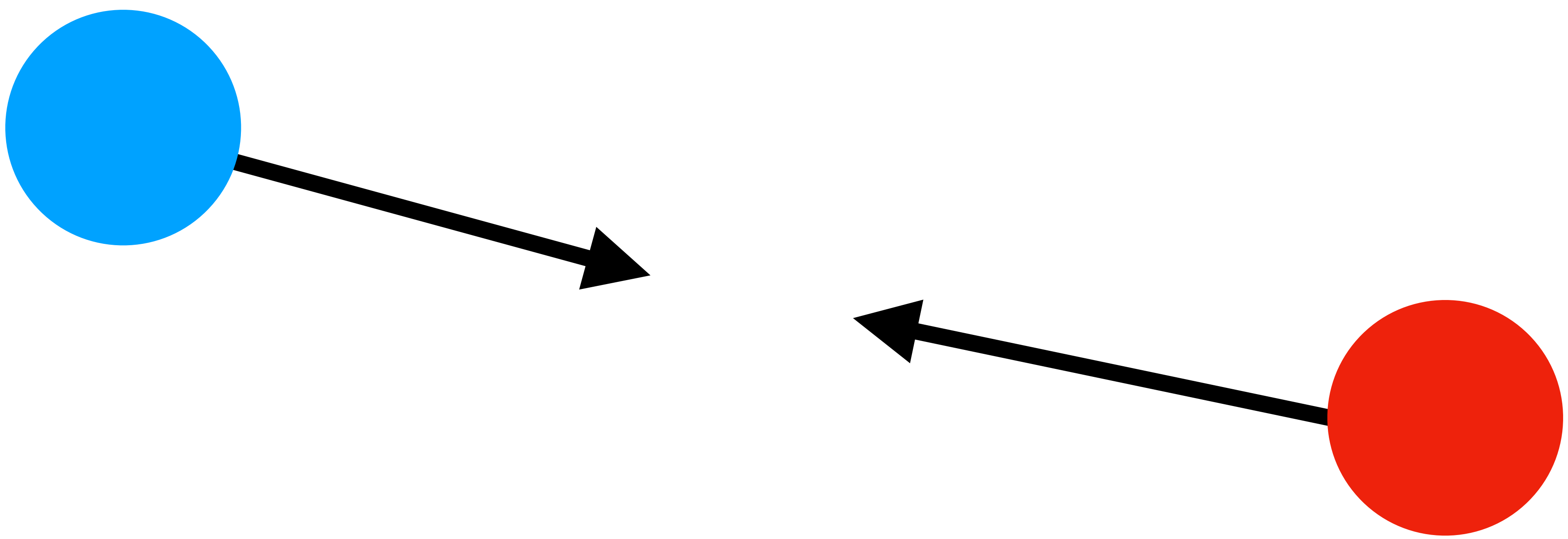
What do we need for Simulation?



$$F = \frac{G m_1 m_2}{|\vec{r}_1 - \vec{r}_2|^3} \left(\vec{r}_1 - \vec{r}_2\right)$$

- All Physics we need has been developed 100s of years ago
    - The two body problem has been solved since the 18th century

What do we need for Simulation?

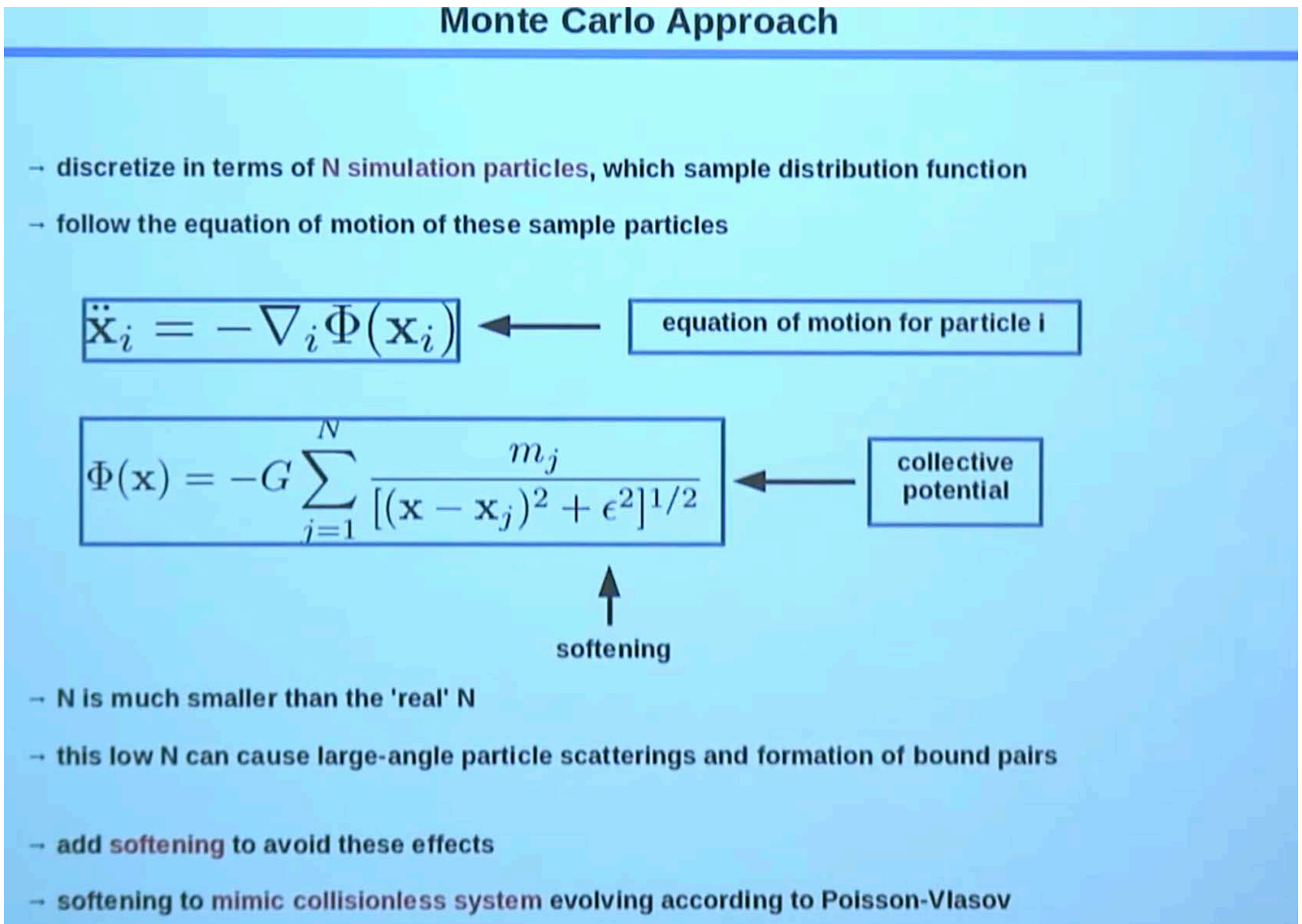


$$F = \frac{G m_1 m_2}{|\vec{r}_1 - \vec{r}_2 + \epsilon|^3} \left(\vec{r}_1 - \vec{r}_2\right)$$

- To simulate this on a computer we will add a “softening” term

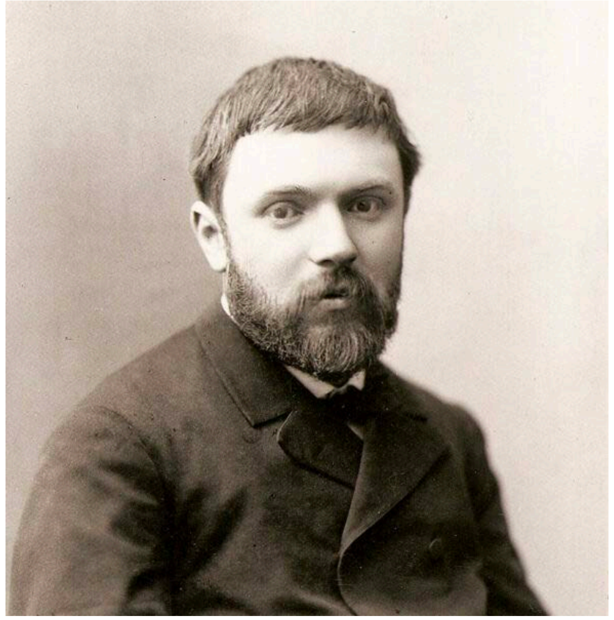
This is how DM/galaxies are modeled

- Instead of treating matter as a fluid
  - Discretize matter into chunks and solve n-body problem



3 body Problem: History

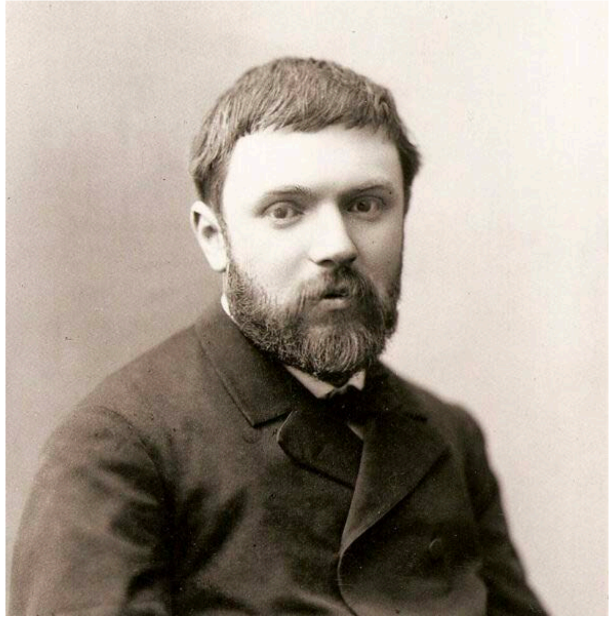
- The original two body probelm was solved in 18th century
  - Work done by Netwon, Bernoulli Bros, Euler, Laplace,…
    - All started on the 3 body problem and built on the two body
- King Oscar II decided to make a competition:
  - For his 60th birthday he bestowed a prize on who could solve the n-body problem



3 body Problem: History

- The original two body probelm was solved in 18th century
  - Work done by Netwon, Bernoulli Bros, Euler, Laplace,…
    - All started on the 3 body problem and built on the two body
- King Oscar II (king of Sweden) decided to make a competition:
  - For his 60th birthday he bestowed a prize on who could solve the n-body problem

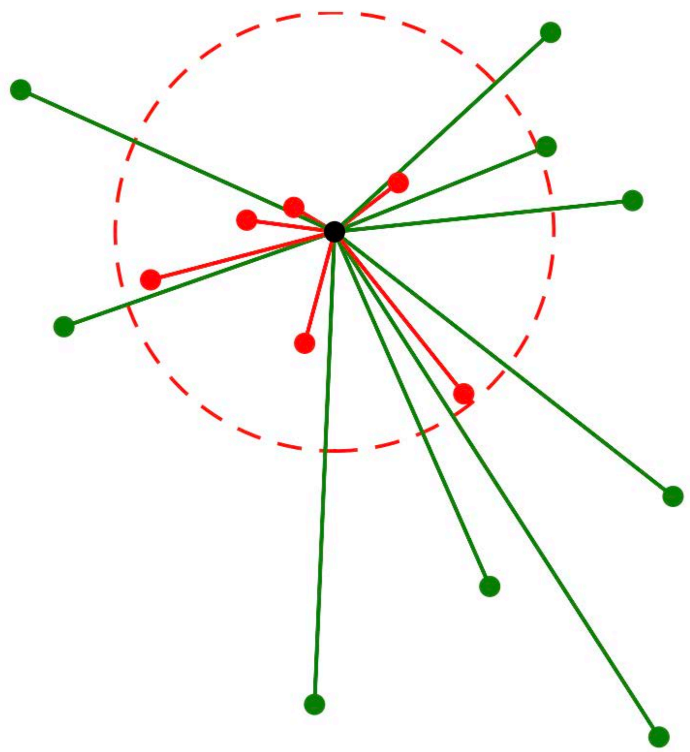
Henri Poincare →



He proved no solution existed!

Going to N-body

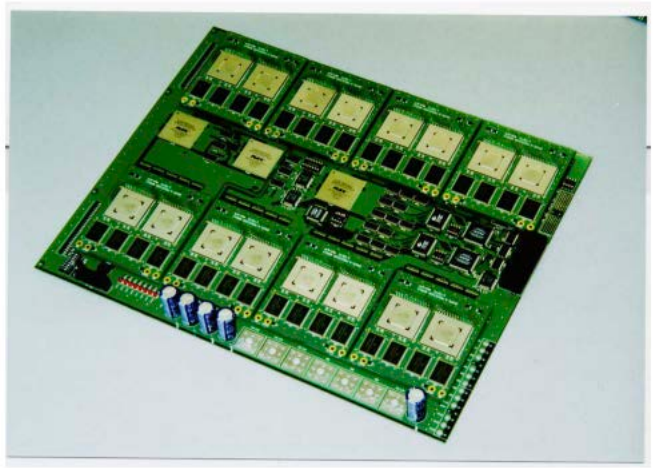
- The challenge of solving this numerically for n-body
  - This scales with the number of bodies N^2
  - Requires the computations of all pairwise distances

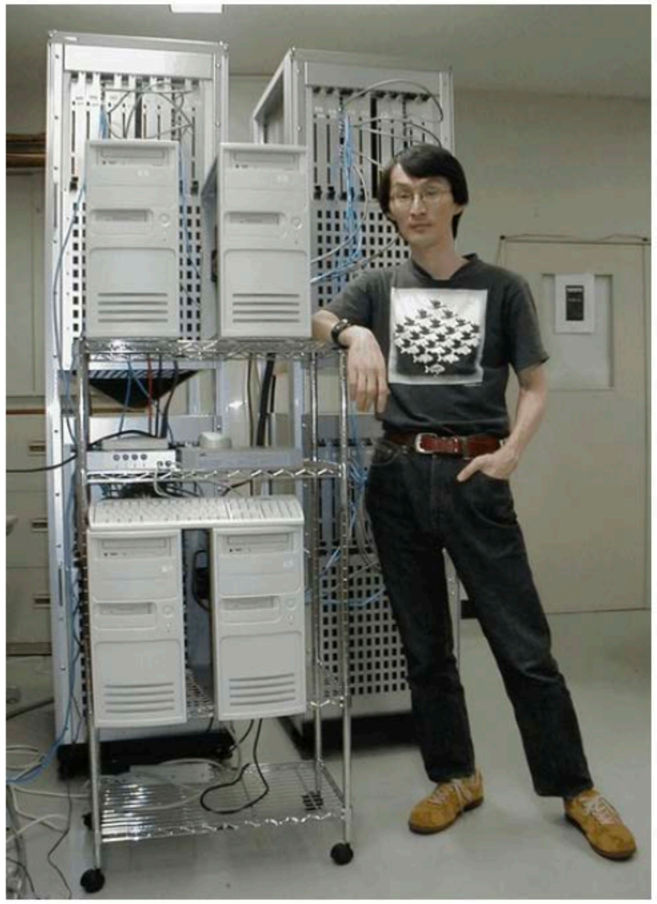


- For N=1000 (1 Million/step computations)
- For $N=10^6$ Something ridiculous

Historical Solutions

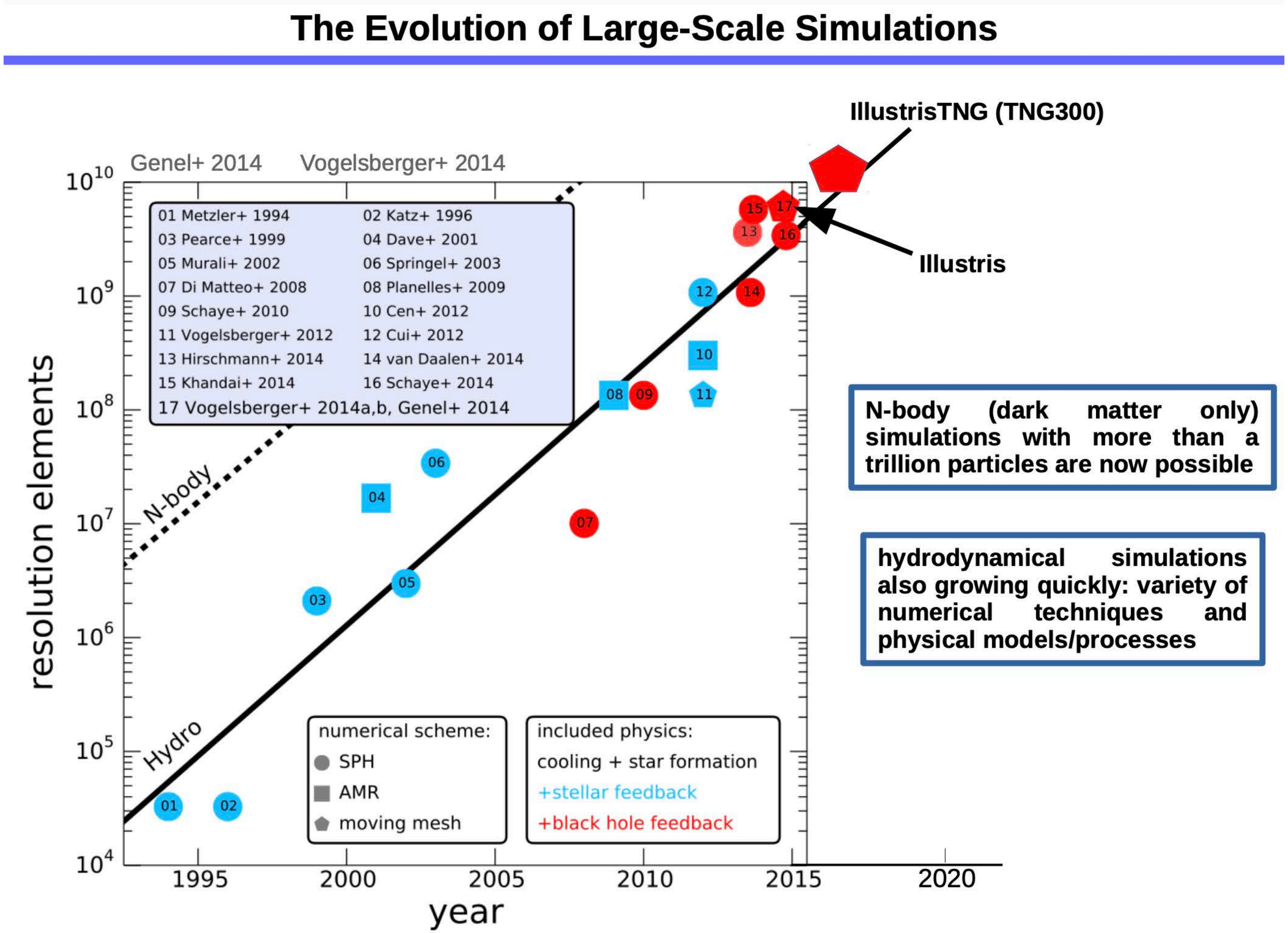
- One approach has been to build dedicated computing hardware
  - Dedicated hardware that can do large-scale parallel computation
  - Focused specifically on n-body simulation
  - GRAPE boards (GRAvity PipelinE)
- Now done with GPUs





Jun Makino with GRAPE-6

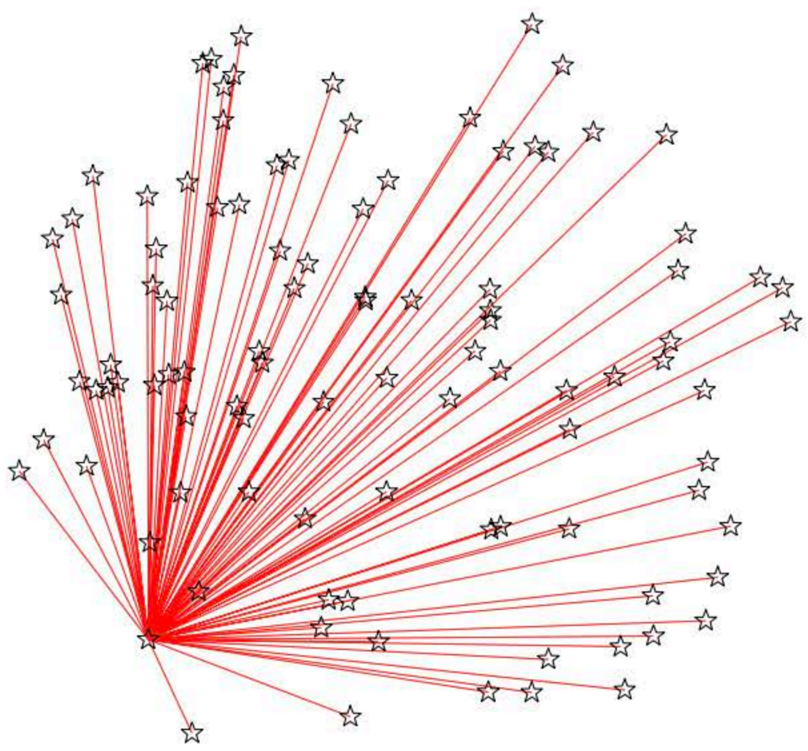
Scaling of n-body

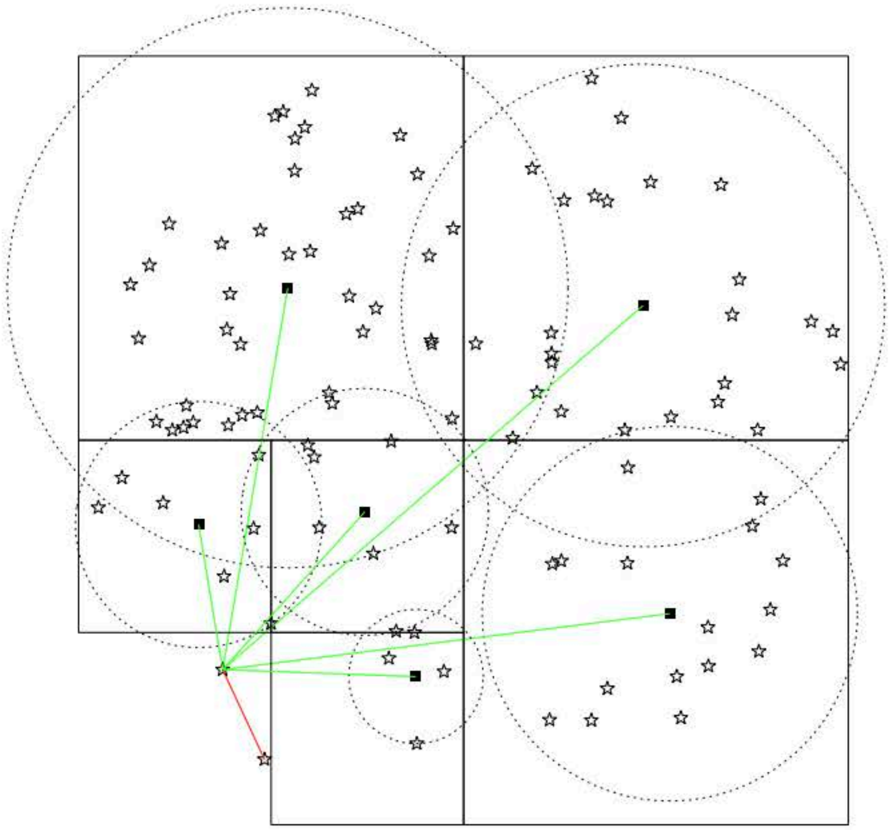


- M. Vogelsberger(https://indico.cern.ch/event/736594/contributions/3184103/attachments/1738225/2812076/talk_vogelsberger.pdf)

How do you deal with N-body?

- Barne's-Hut Algorithm re-rank stars into a tree structure
    - Structure is a grid over the whole space

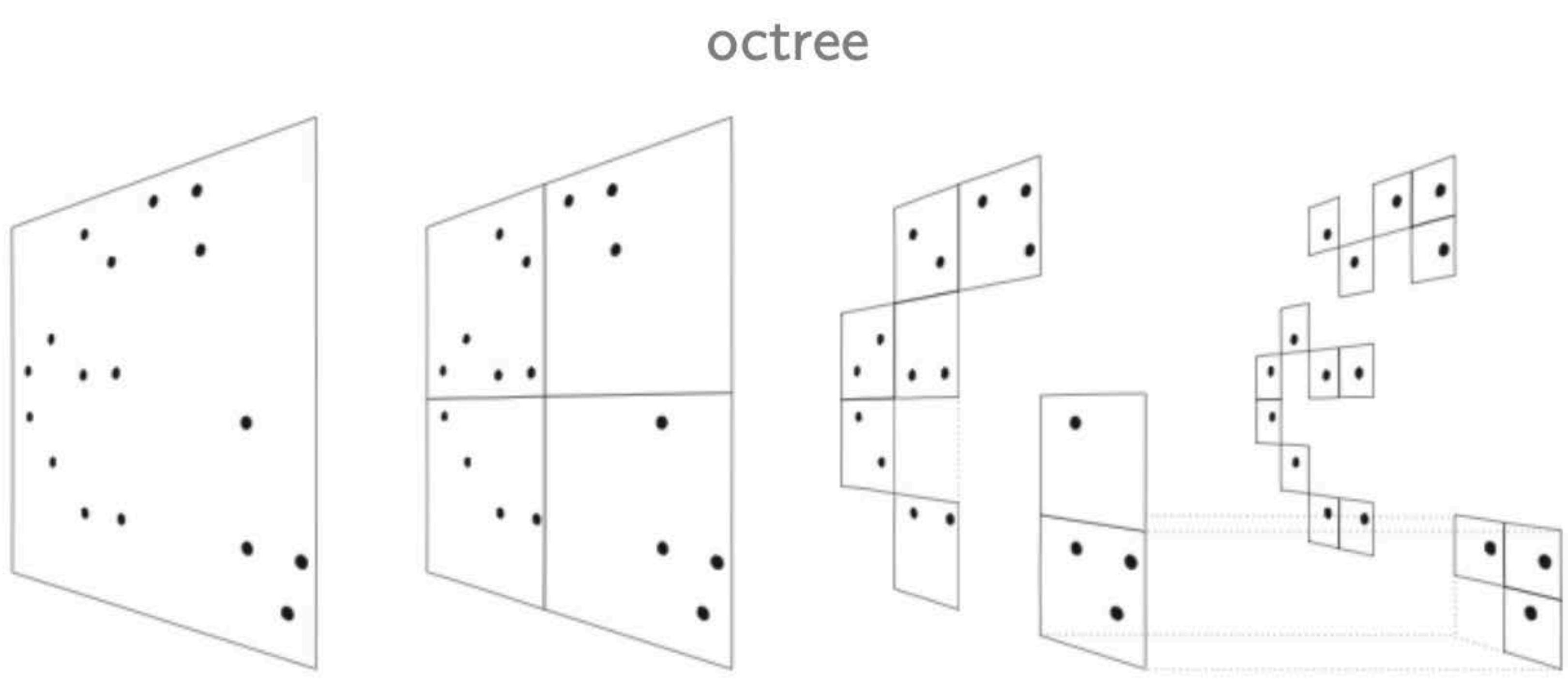


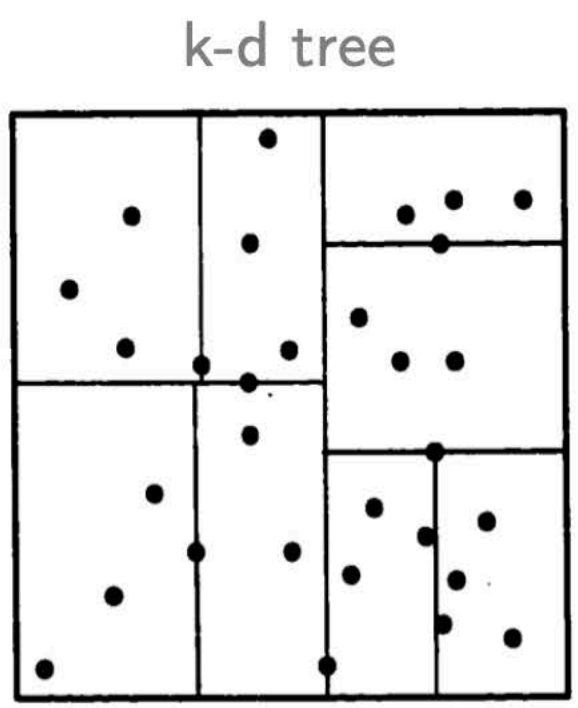


[from Dehnen & Read 2011]

Tree Construction

- QuadTree/OctTree
  - Split each square(cube) into 4(8) sub regions
- KD Tree
  - Use the data to draw equal numbered regions in space

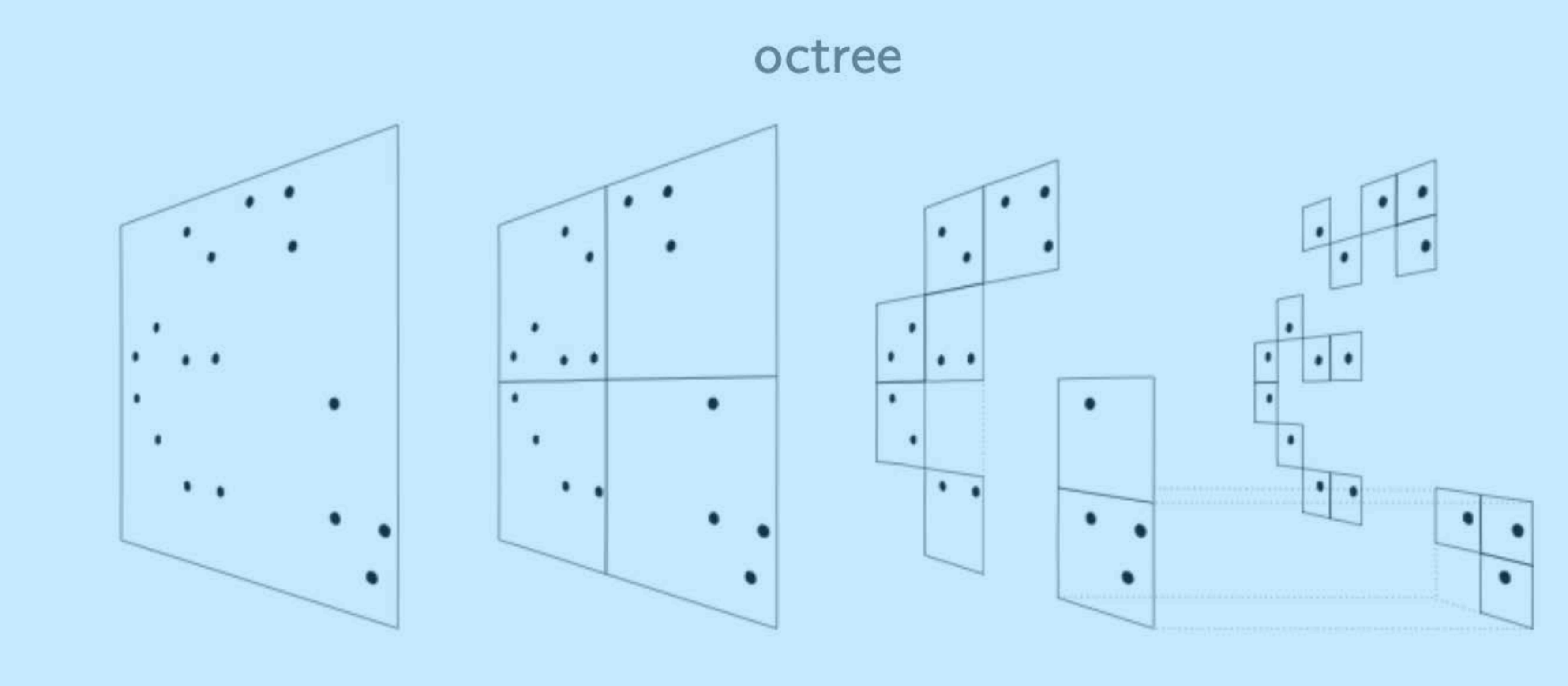


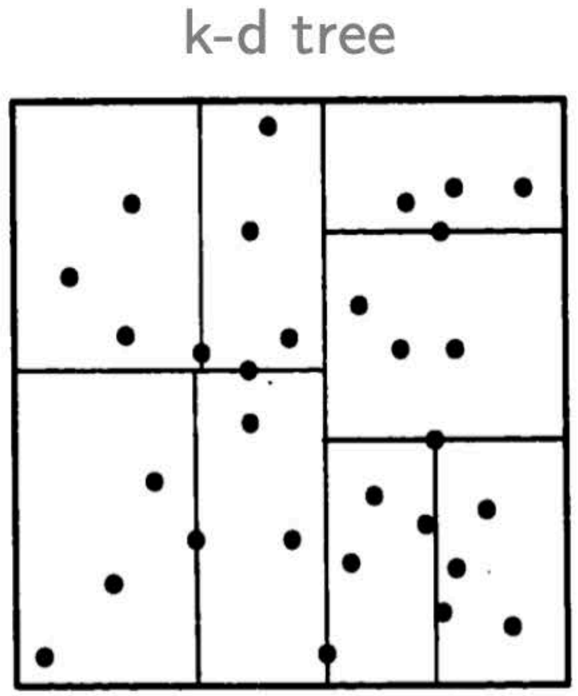


Tree Construction

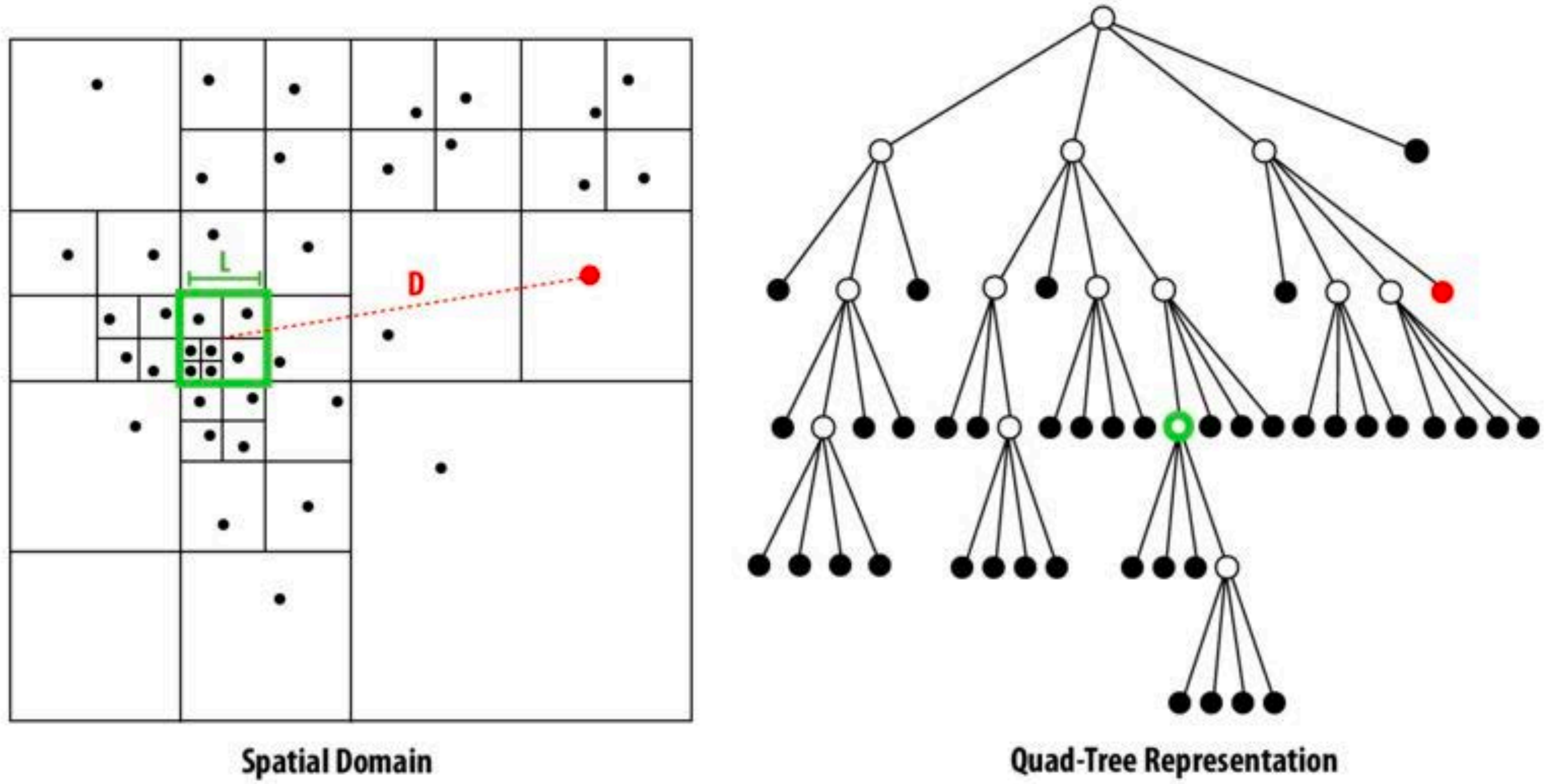
- QuadTree/OctTree
  - Split each square(cube) into 4(8) sub regions
- KD Tree
  - Use the data to draw equal numbered regions in space

Just requires that we know the bounds of the space





Visualizing Tree



Note that for the big grids, this equates to Gauss' law style an apporach
Treat each square as a star w/total mass at mass weighted center

Barnes-Hut Algorithm

- We can follow a step by step construction of this :
  - 1. Construct tree structure with bounds
  - 2. Loop over stars and fill tree structure
  - 3. Loop over stars and compute distance
    - Full n-body computation for nearby trees only
  - 4. Step forward everything
- The above process is N log(N) in comptuational time

Image Sources

image

link:

attribution: