

Virtual universes vs. the real thing

Andreea Font, Juliana Kwan, Ian McCarthy, Jaime Salcido
(on behalf of the Computational Galaxy Formation group)

Astrophysics Research Institute,
Liverpool JMU



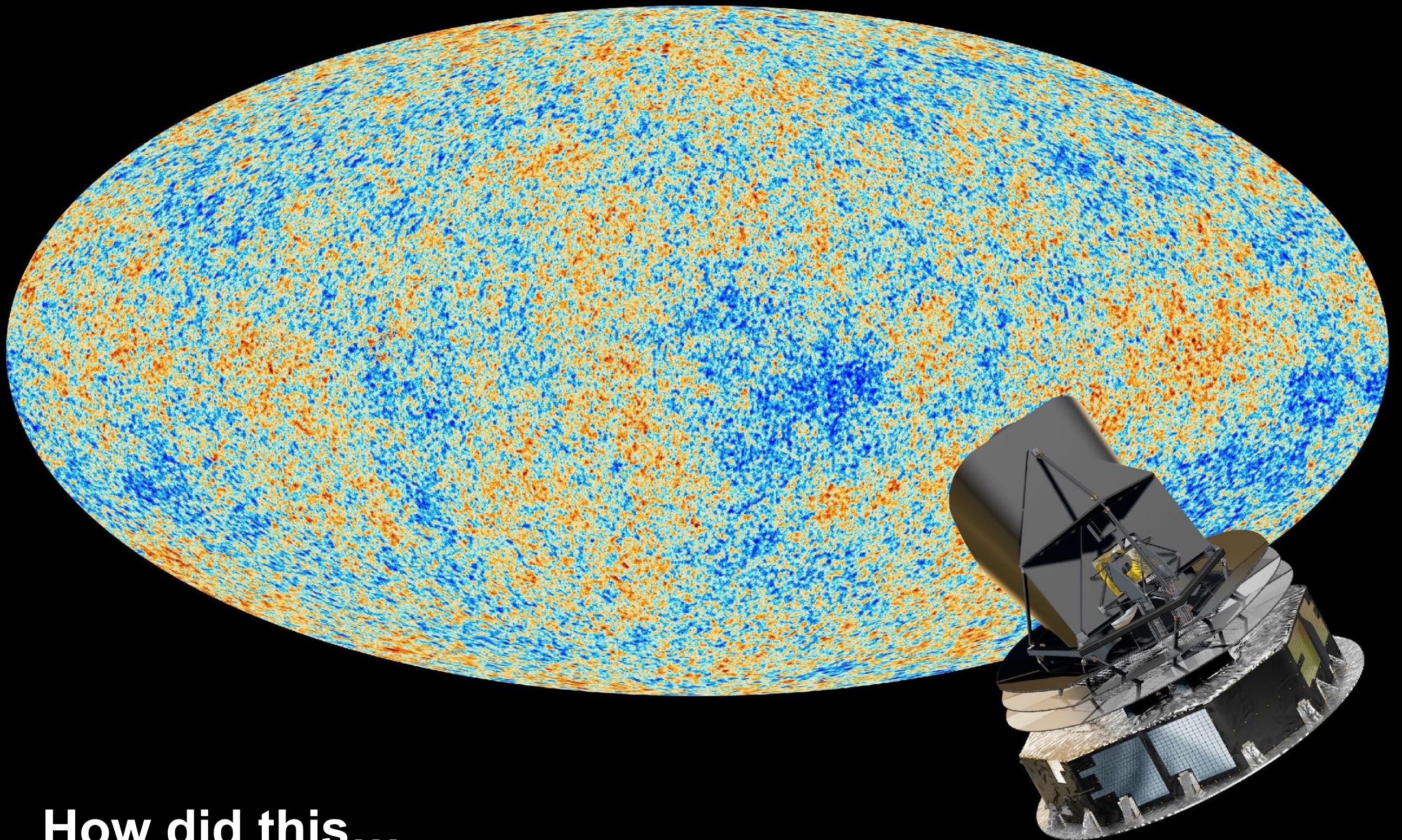
Outline

- Cosmological simulations (“virtual universes”) - Ian
- The EAGLE database & visualisation - Jaime
- The Milky Way & Gaia - Andreea
- Parameter inference & MCMC - Juliana

Virtual universes

Ian McCarthy
(with thanks to Rob Crain)

14 billion years ago, the Universe looked like this.



How did this...

...evolve into this?

M87



M82



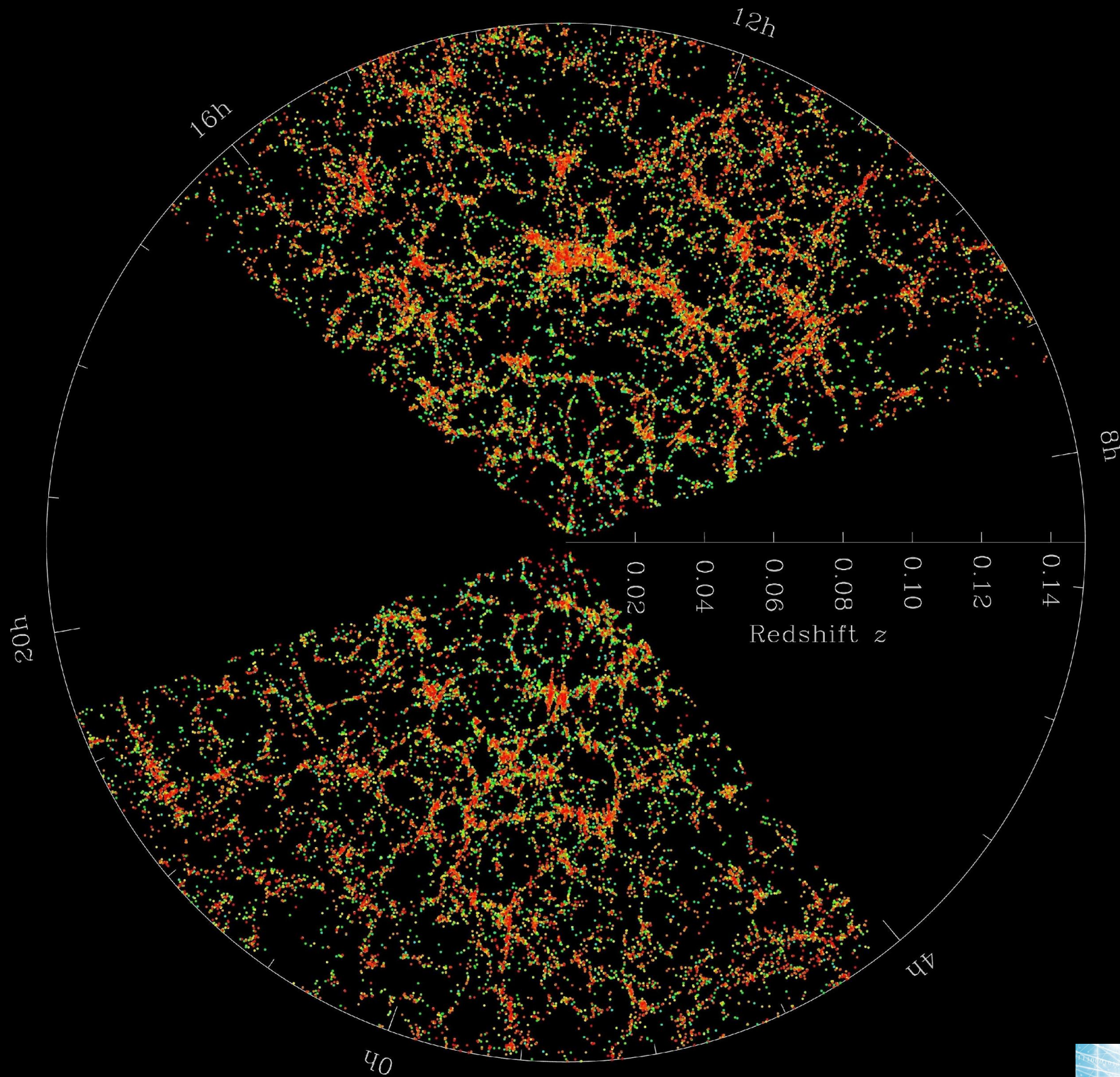
GN-z11



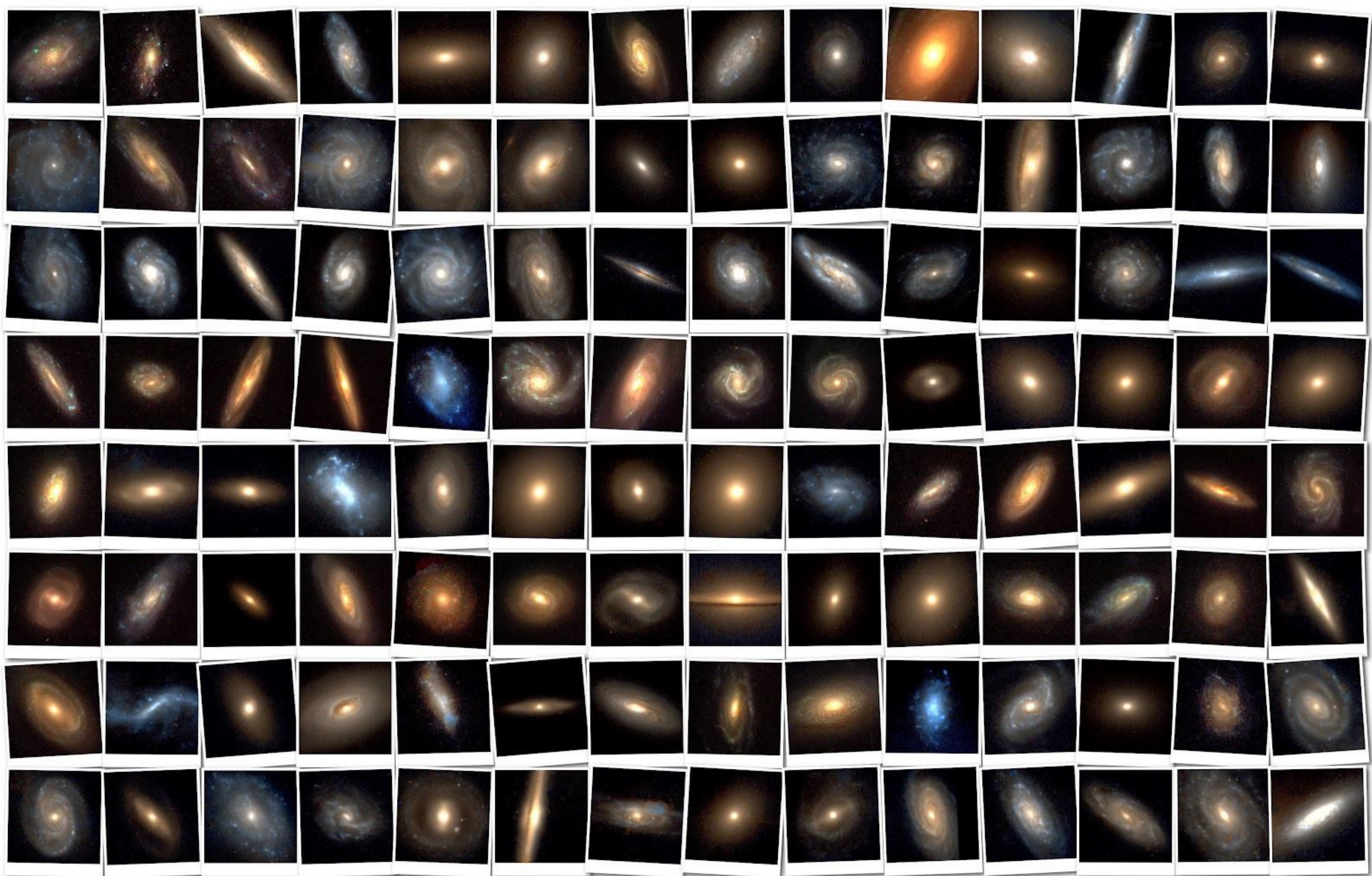
The formation & evolution of the galaxy population:

- How do dark matter, gas, stars and black holes assemble into galaxies?
- How do galaxies populate the cosmic large scale structure?
- How are galaxies influenced by physical processes & their environments?

Key question: why this distribution of galaxies?



Key question: why this diversity of galaxies?



**Key question: what drives these
apparent environmental effects?**

**Unlike most scientists,
astrophysicists cannot
learn by performing
experiments...we need
'digital laboratories'.**

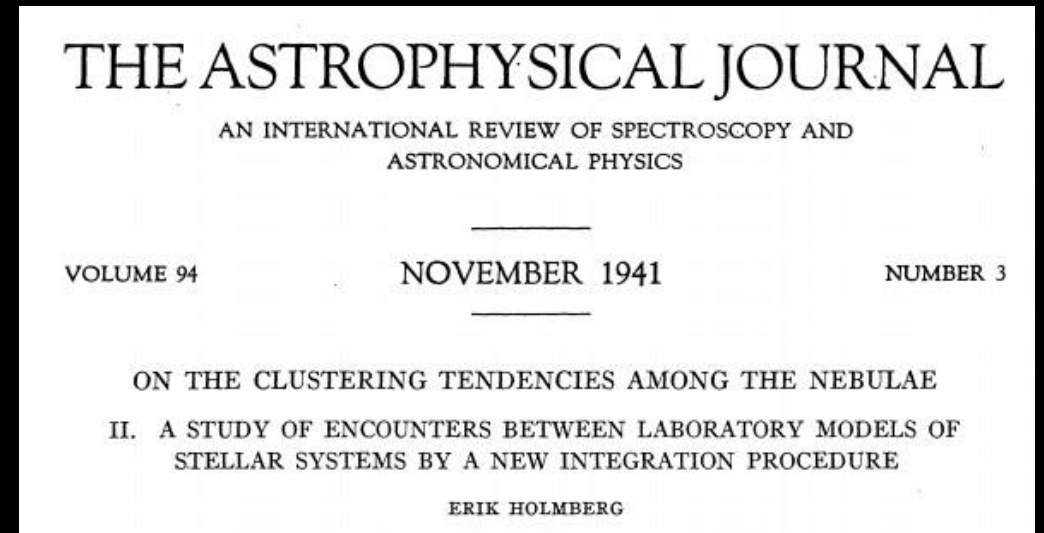


First attempt with analogue computers: Holmberg (1941)

Simulate 2 galaxies each with 37 “stars”, so 74 stars.

Each interacts with 73 others; total # of interactions per step = $73 \times 74 = 5402$

No fun with pencil and paper: use lightbulbs and light meters.



Feed initial conditions into a computer

(The Universe shortly after Big Bang is remarkably well understood)



Program the computer with descriptions of the key astrophysical processes

(These are understood in vastly differing degrees)



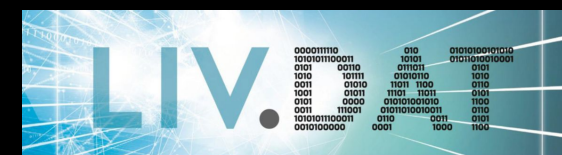
Evolve the initial conditions for 14 billion years, from the Big Bang to the present day

(Requires lots of calculations, i.e. a big computer)



Compare with observational data (or make predictions)

(Occasionally we show the observers got it wrong!)



The ultimate calculation?

- Length scales

- Event horizon of galactic black hole: ~ 100 Earth orbits: 10^{13} m
 - Size of observable Universe: ~ 10^{10} light years: 10^{28} m
- 15 orders of magnitude

- Mass scales

- Mass of star clusters: ~ 10^4 solar masses 10^{34} kg
 - Mass within obs. Universe: ~ 10^{21} solar masses 10^{51} kg
- 17 orders of magnitude

- Time scales

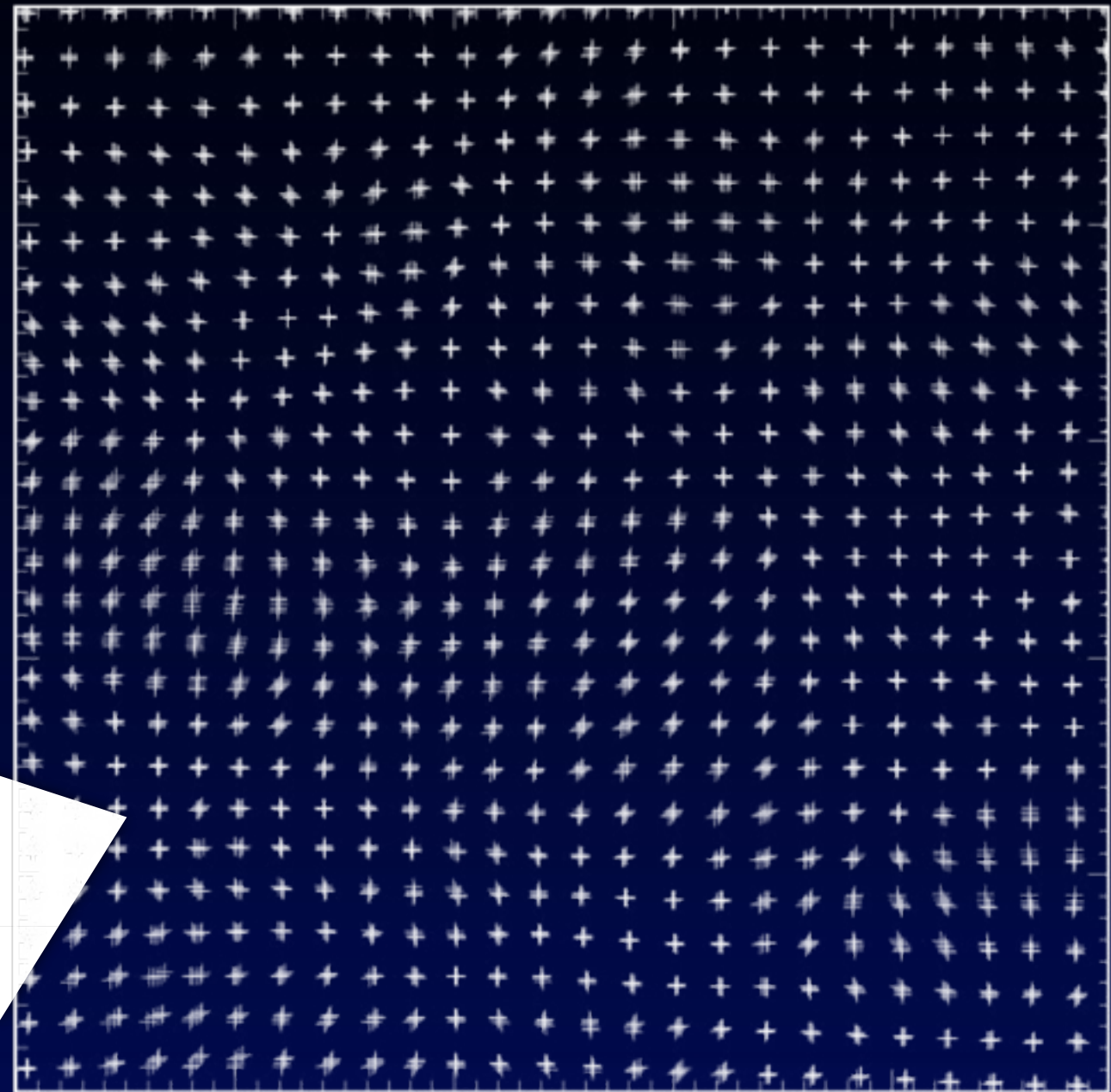
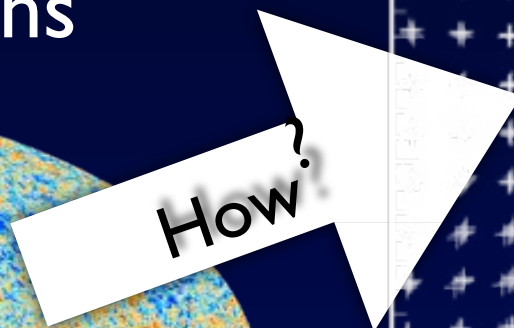
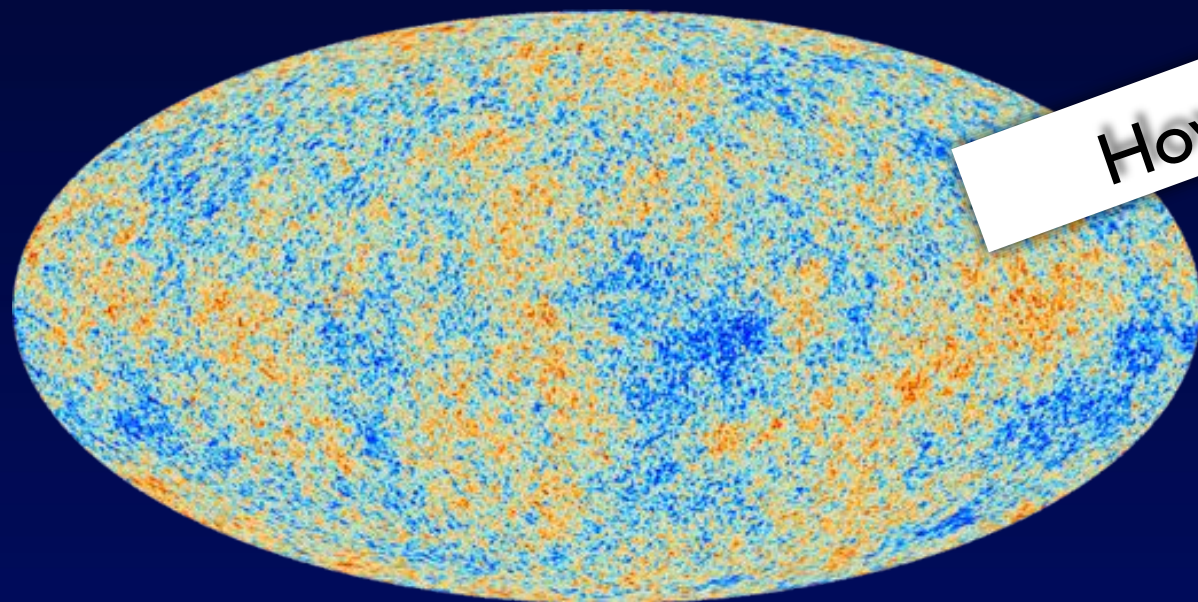
- Variability of galactic black hole: ~ few days 10^5 s
 - Age of the Universe: ~14 billion years 10^{17} s
- 12 orders of magnitude

This would require a computer with 10^{12} processors and 10^9 TB of memory. It would need 100 million times the current global electricity consumption. It would run for 10 million years (at least)!



Perturbations

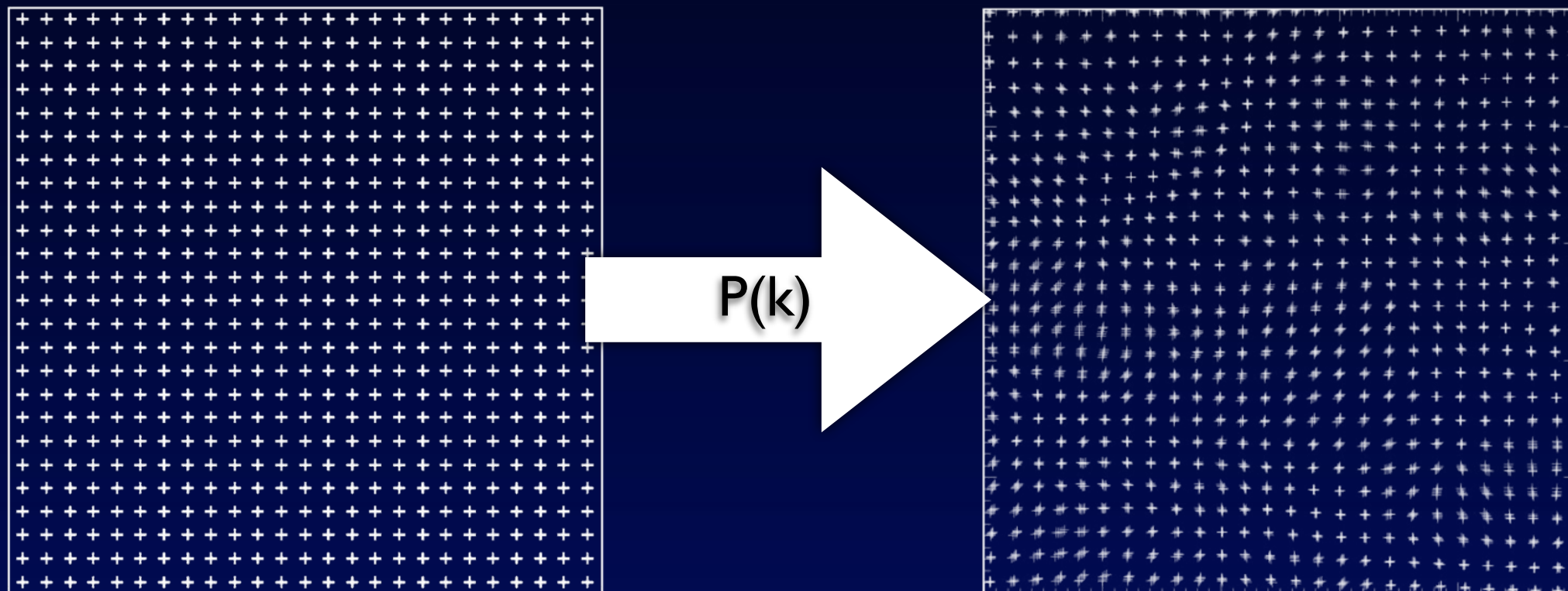
- Universe shows small density fluctuations already at high z (see CMB)
- Convert CMB fluctuations to density perturbations



perturbed density field

The power spectrum

- Use the power spectrum to describe the density fluctuations
- From inflation: $P_i(k) = Ak^n$
- Temporal evolution (in the linear regime): $P(k, t) = P_0(k)D^2(t)$



Power spectrum tells us “how far” to displace particles from their starting positions. Fluctuations were a Gaussian random field - direction of perturbation is random.

Limitations of this method

- There are a number of parameters that have to be chosen:

- ▶ Box size B

- ▶ Number of particles N

- ▶ Starting redshift z_i

- In practice there are several constraints on these:

- ▶ Minimal modes that are included: $2B/\sqrt[3]{N}$

- ▶ Largest mode has to stay linear: $B \geq 2\pi/k_{nl} \sim 20Mpc$

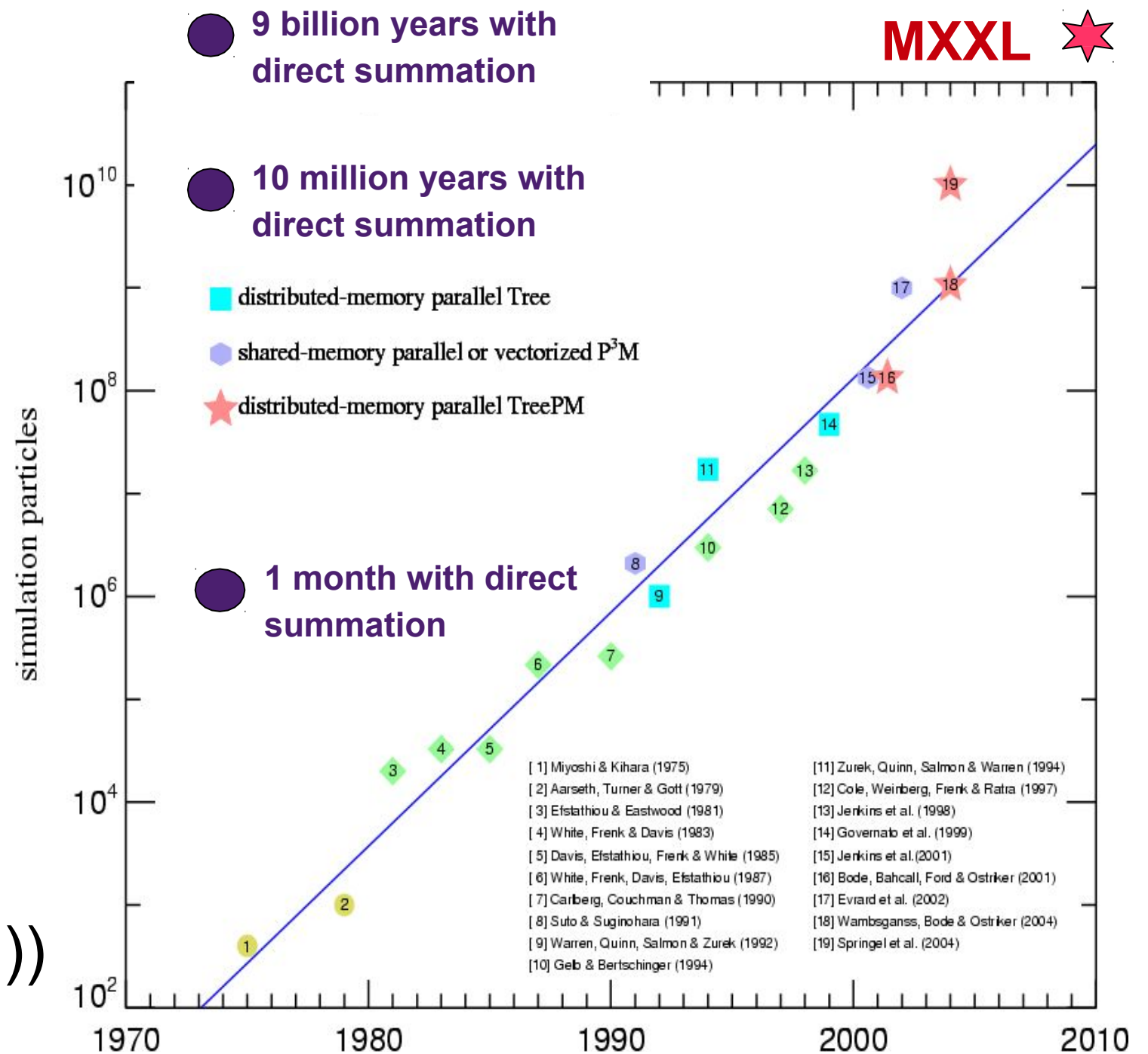
- ▶ Starting redshift (typically $40 < z_i < 80$):

Too late: shell crossing not taken into account

Too early: numerical noise is integrated

Size of cosmological simulations over time

- computers double speed every 18 months (Moore's law)
- particle number in simulations doubles every 16-17 months
- only possible with algorithms that scale close to $\sim N$ (or $N \log(N)$)



**Poisson's equation:
potential due to a
"charge" e.g. mass**

The particle-mesh (PM) method

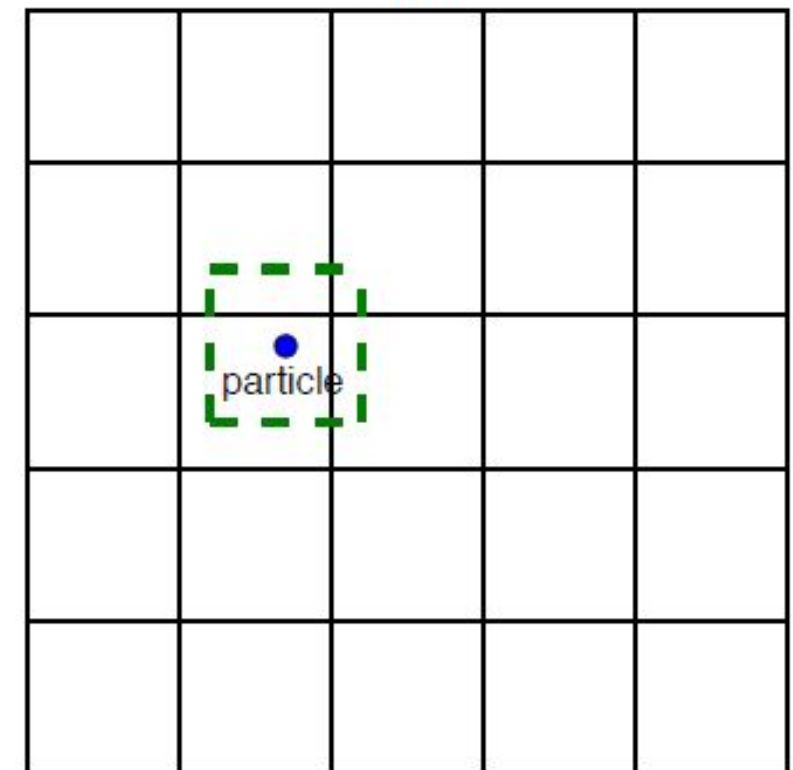
- particle-mesh method
 - Poisson's equation in real space:
 - Poisson's equation in Fourier space:
 - assign particle mass to grid (e.g. CIC)
 - compute Fourier transform of density contrast (FFT $\sim N \log N$)
 - convert to Fourier transform of potential
 - transform the potential back to real space
 - compute gradient by finite differencing of the potential

average comoving matter density $\bar{\rho}_c$

$$\vec{\nabla}^2 \delta\phi = 4\pi G \bar{\rho}_c \delta a^{-1}$$

$$-k^2 \delta\phi_{\vec{k}} = \frac{4\pi G \bar{\rho}_c \delta_{\vec{k}}}{a}$$

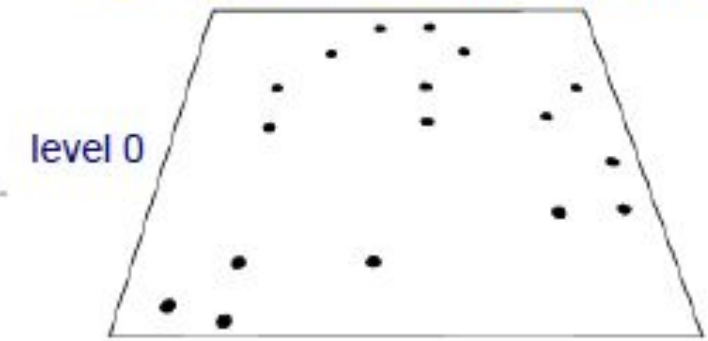
mesh



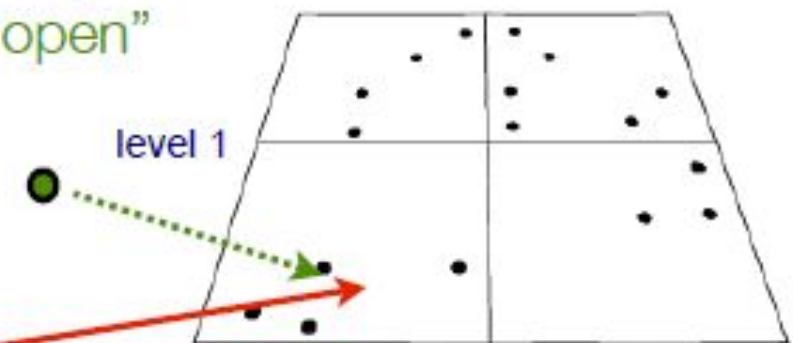
The tree method

- tree algorithm:
 - group distant particles together and use their multipole expansion
 - only $\sim \log(N)$ force terms per particle

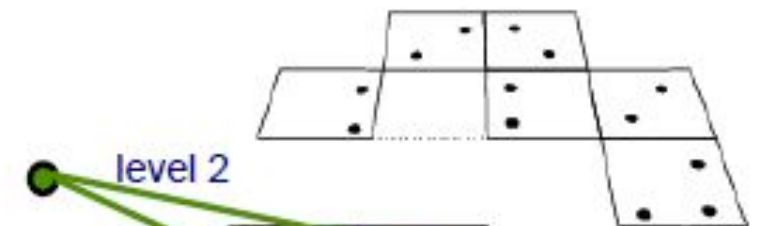
Oct-tree in two dimensions



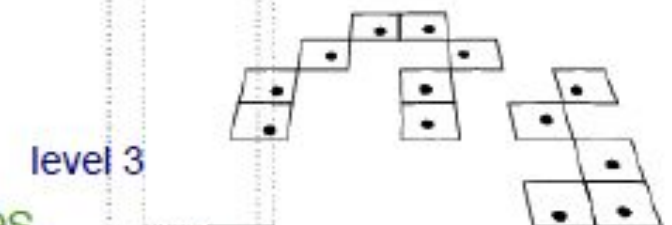
for force exerted by nearby particles "open" node



for force exerted by distant particles use coarse node

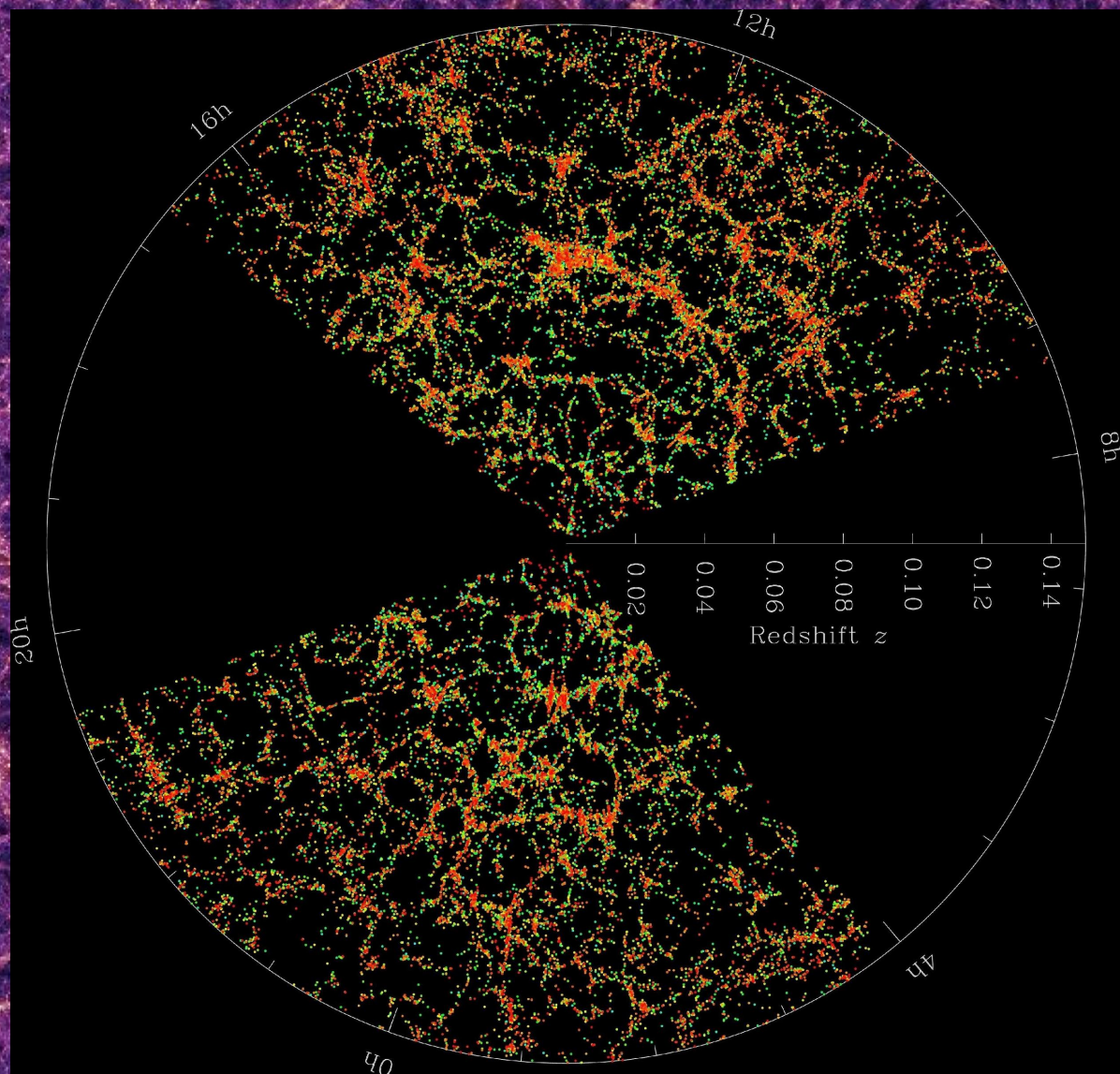
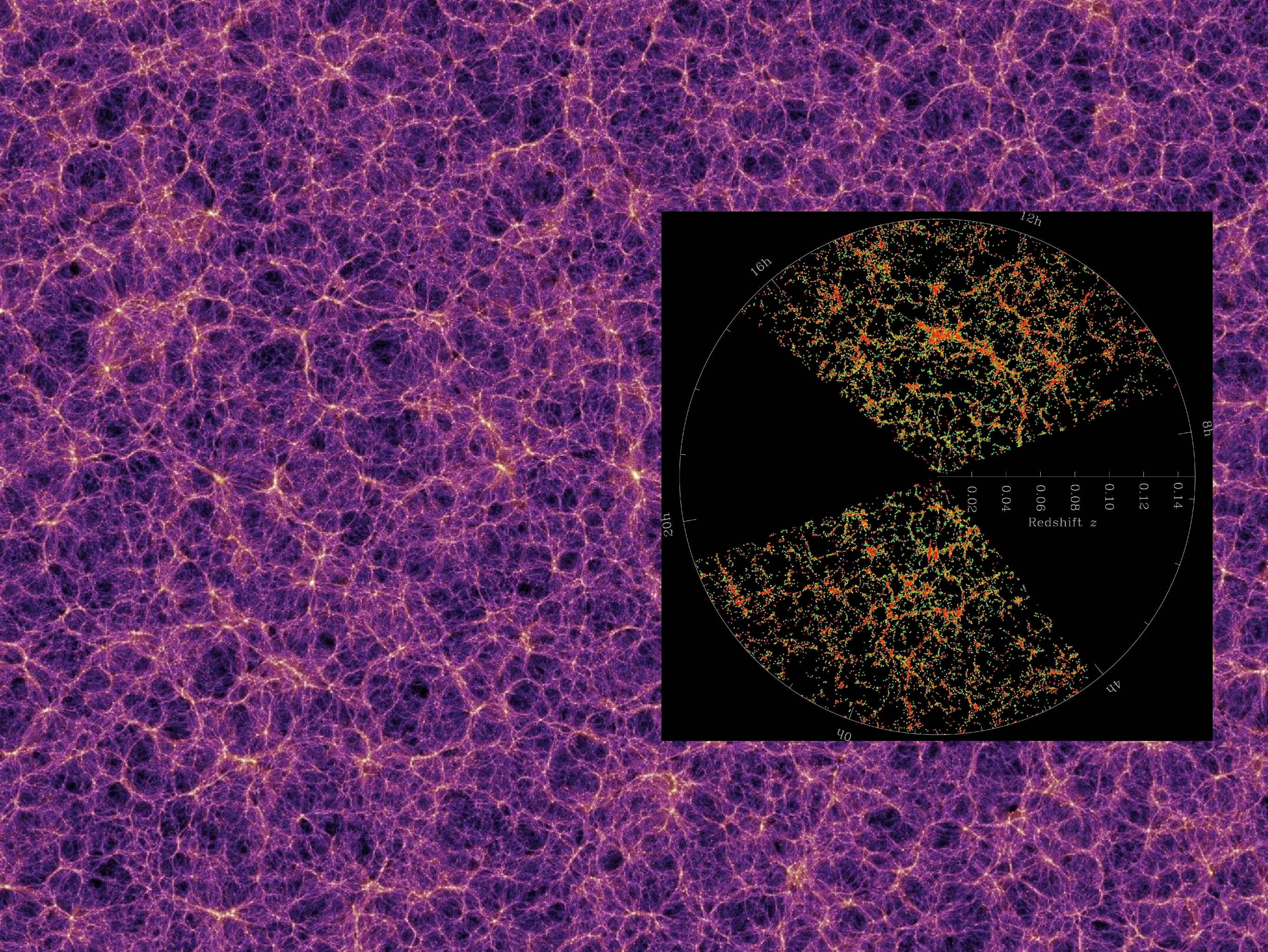


and compute force based on sub-nodes



or individual particles

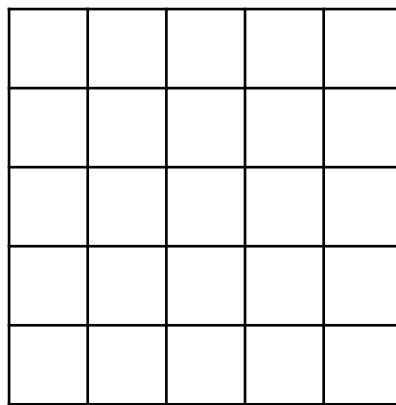
original image: Springel et al. 2001



Eulerian vs Lagrangian methods

Eulerian methods moving-mesh

discretize space
(finite-volume scheme)



use a grid fixed in space

discretize space
(finite-volume scheme)

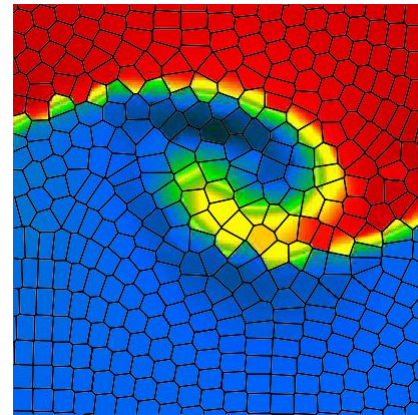
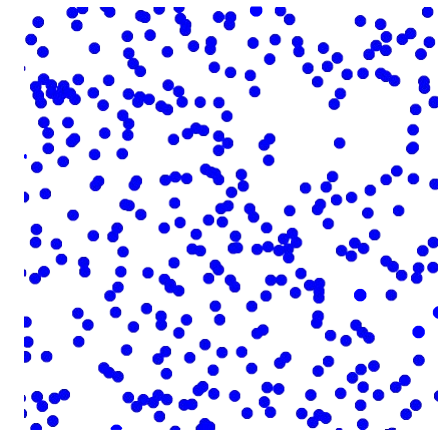


image credit: V.
Springel

uses an unstructured
mesh moving with the flow

Lagrangian methods

discretize mass



use particles for the gas
(like in n-body) which
move with the flow

What other ingredients are necessary?

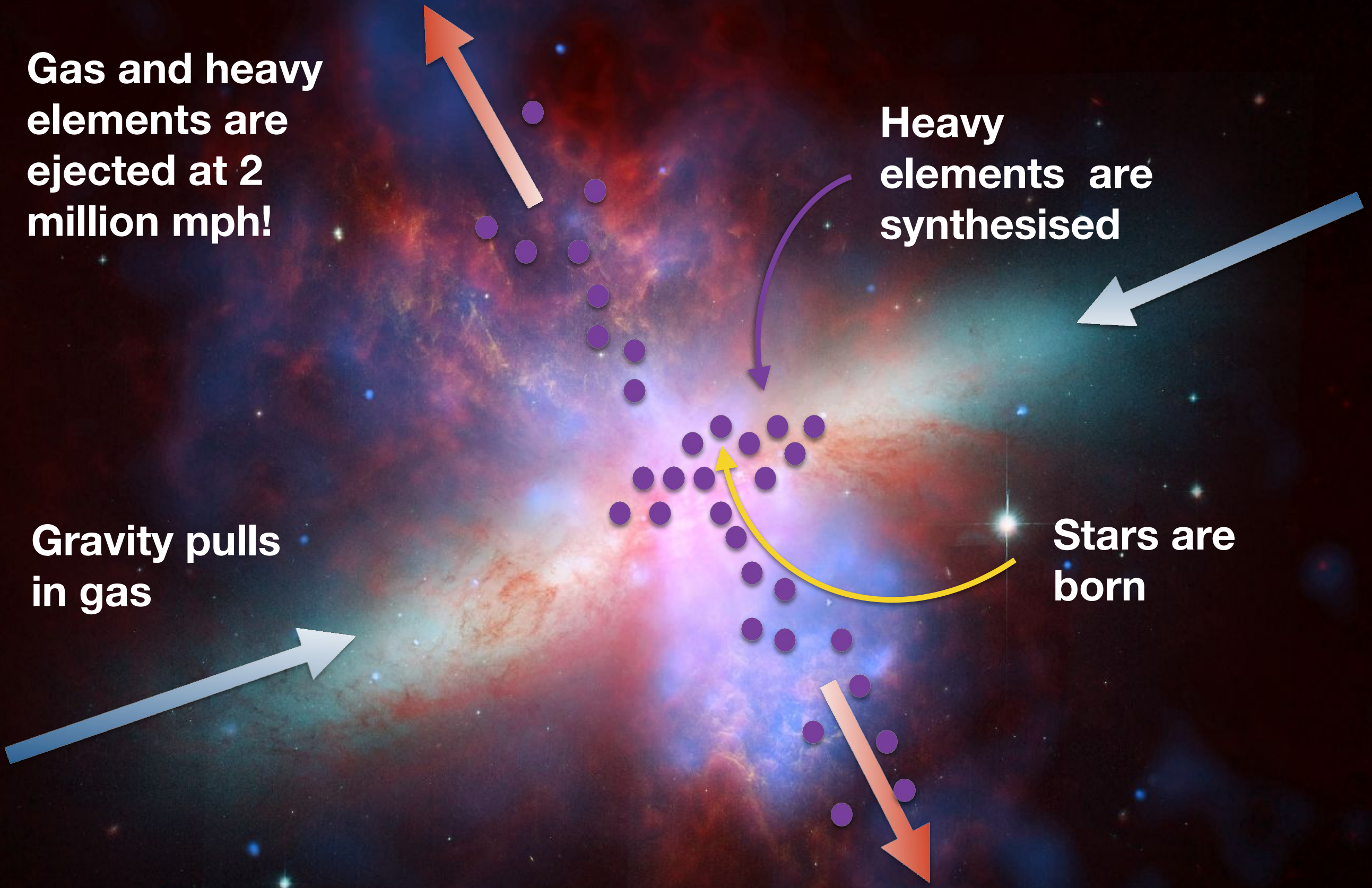


Gas and heavy elements are ejected at 2 million mph!

Heavy elements are synthesised

Gravity pulls in gas

Stars are born



Gas and heavy elements are ejected at 2 million mph!

Gravity pulls in gas

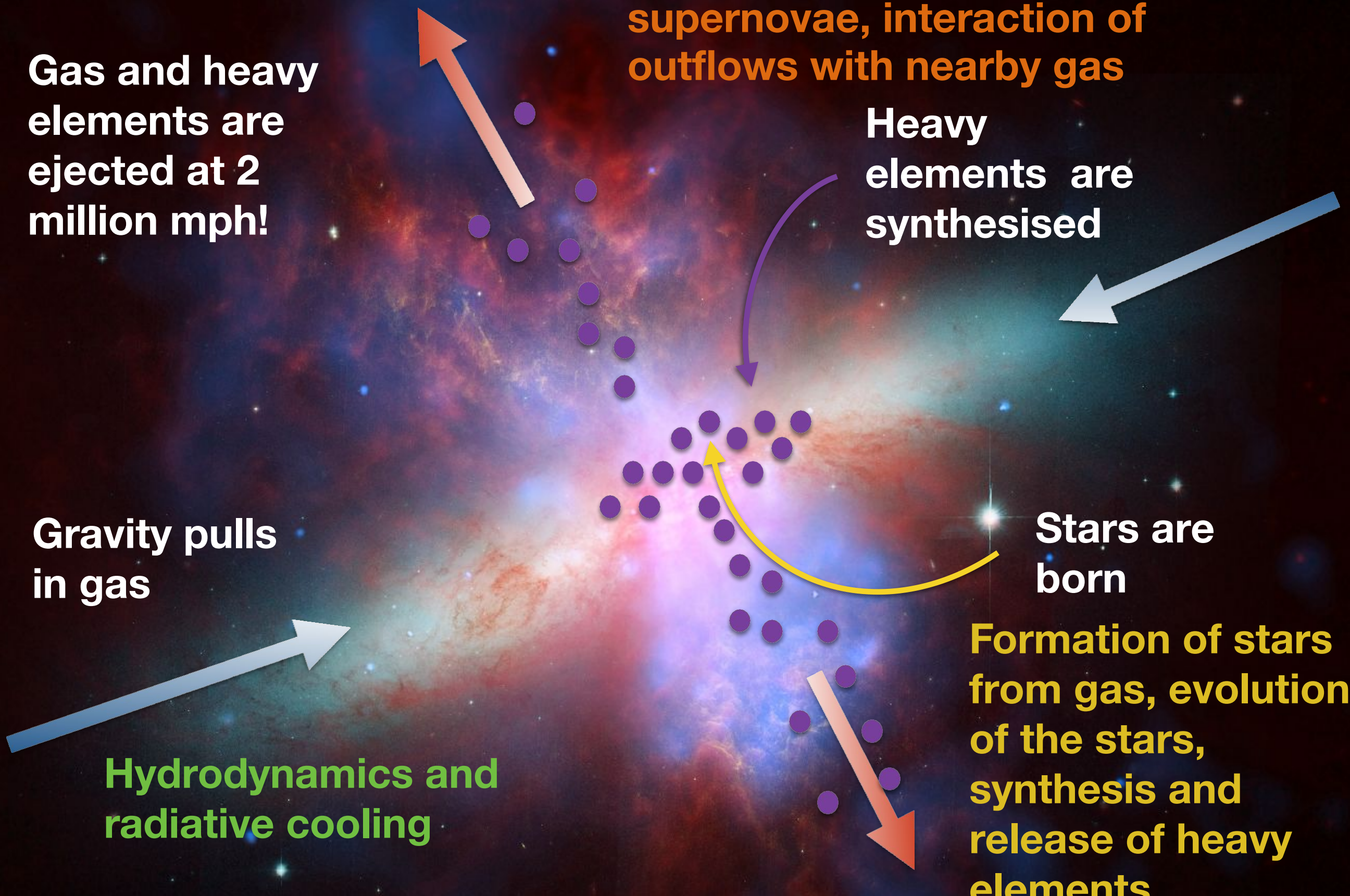
Hydrodynamics and radiative cooling

Injection of energy from supernovae, interaction of outflows with nearby gas

Heavy elements are synthesised

Stars are born

Formation of stars from gas, evolution of the stars, synthesis and release of heavy elements



What other ingredients are necessary?

Gravity (all components)

Hydrodynamics for compressible fluid

Everything else...is below our resolution limit!

Radiative cooling & heating of gas

Star formation

Stellar evolution & nucleosynthesis

Black hole formation and growth

Feedback from stars and black holes

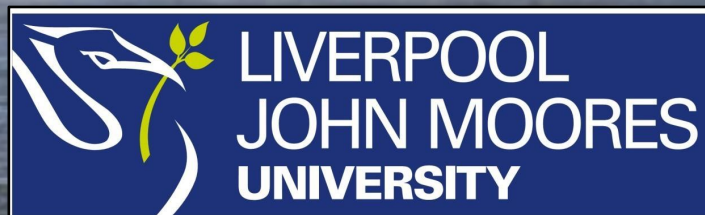
Intimately coupled in a non-linear fashion!

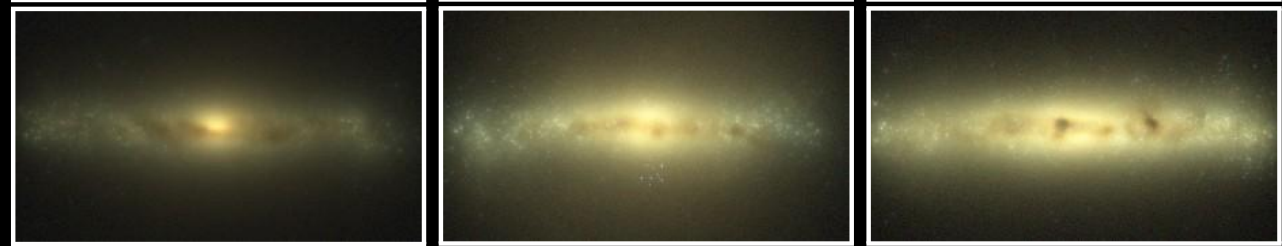
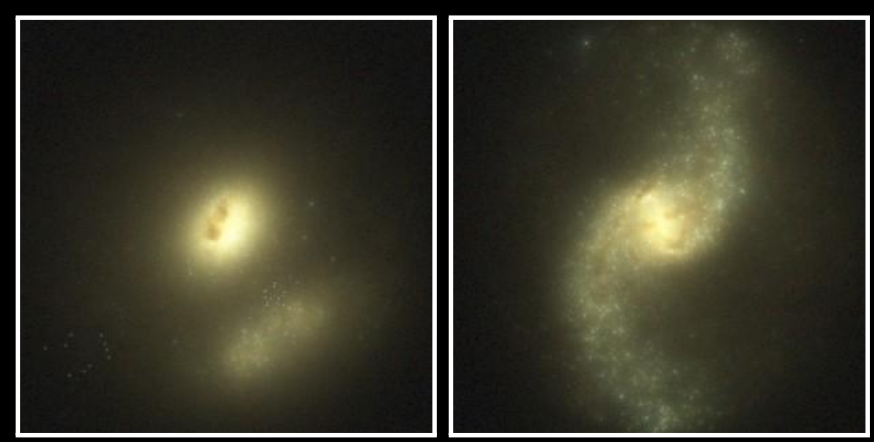
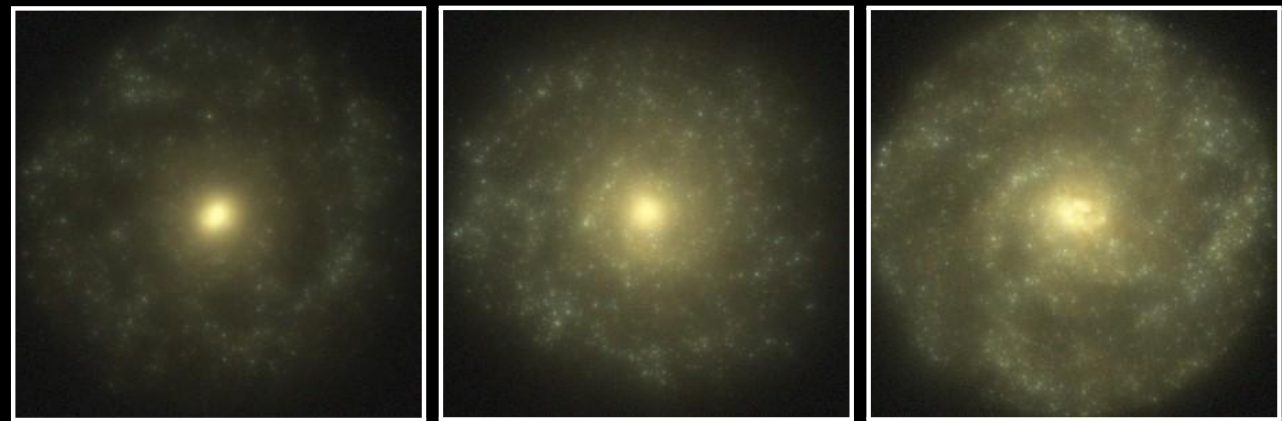
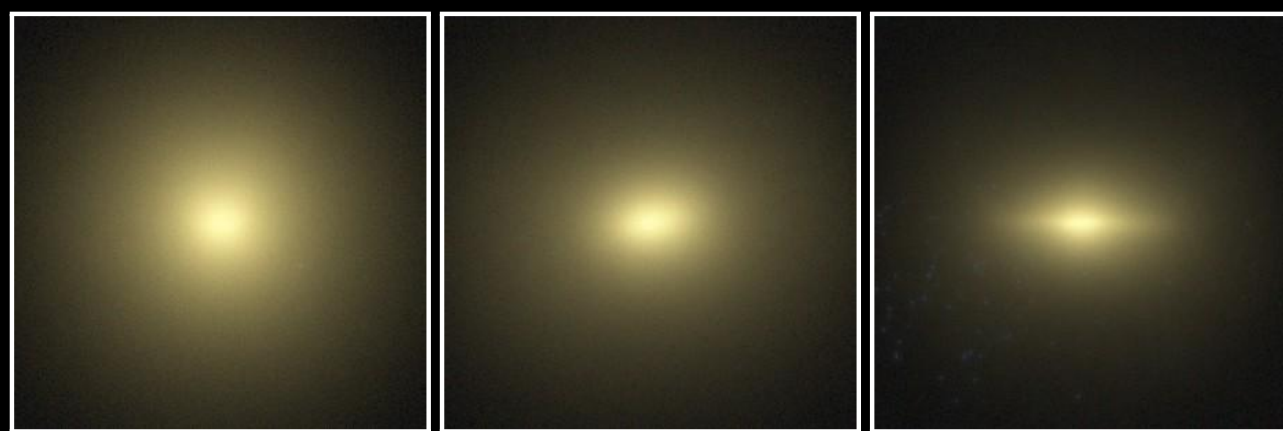
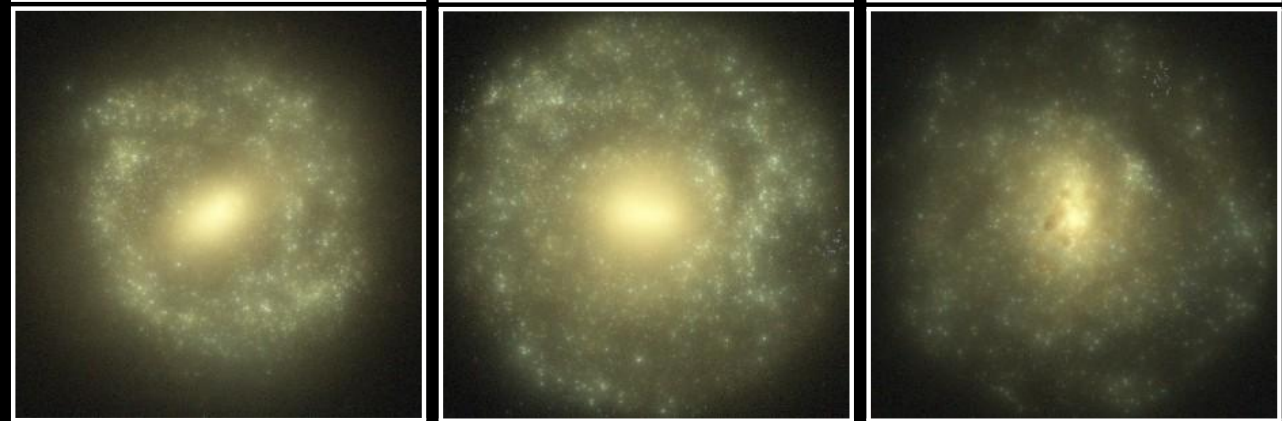
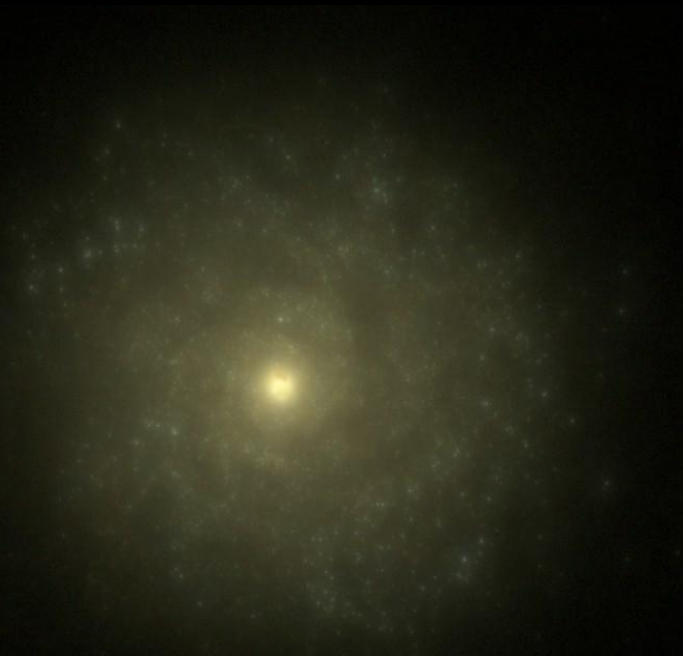




EAGLE

Evolution and Assembly of GaLaxies and their Environments





Summary

HPC central to building an understanding of how the early Universe evolved into that we observe today.

Confrontation of dark matter simulations with observed galaxy distribution is major evidence for a CDM-dominated Universe.

Realism of simulations including complex “gastrophysics” has broadly corroborated galaxy formation theory in CDM cosmogony.

Challenge used to be “can we reproduce what we see?”. But now, sims are the premier tool with which to interpret observations. Simulations give us access to components (and epochs!) that we cannot easily observe.

Still huge amount of “known unknowns” to tackle! Much of the gastrophysics is implemented phenomenologically: we want to understand the physics of the messy stuff in detail!

Questions?

Eagle database & PySPHviewer

Jaime Salcido

The EAGLE public database contains galaxy properties (such as masses, star formation rates, luminosities and metallicities), merger histories and images for more than 1,000,000 simulated galaxies extracted from multiple simulations of various box sizes, numerical resolutions and physical models.

You can access the database in the following link:

<http://icc.dur.ac.uk/Eagle/database.php>

Useful information about the simulations and available datasets can be found in the data release paper:

<https://arxiv.org/abs/1510.01320>

EAGLE Database

Documentation ← 9)

CREDITS/Acknowledgments ← 10)

Public Databases

▢ Eagle

▢ Tables

AGNdT9L0050N0752_Aperture
AGNdT9L0050N0752_FOF
AGNdT9L0050N0752_Magnitudes
AGNdT9L0050N0752_Sizes
AGNdT9L0050N0752_SubHalo
RecalL0025N0752_Aperture
RecalL0025N0752_FOF
RecalL0025N0752_Magnitudes
RecalL0025N0752_Sizes
RecalL0025N0752_SubHalo
RefL0025N0376_Aperture
RefL0025N0376_FOF
RefL0025N0376_Magnitudes
RefL0025N0376_Sizes
RefL0025N0376_SubHalo
RefL0025N0752_Aperture
RefL0025N0752_FOF
RefL0025N0752_Magnitudes
RefL0025N0752_Sizes
RefL0025N0752_SubHalo
RefL0050N0752_Aperture
RefL0050N0752_FOF
RefL0050N0752_Magnitudes
RefL0050N0752_Sizes
RefL0050N0752_SubHalo
RefL0100N1504_Aperture
RefL0100N1504_FOF
RefL0100N1504_Magnitudes
RefL0100N1504_Sizes
RefL0100N1504_SubHalo
Snapshots

6) Available Simulations

Welcome <Your User Name>

Streaming queries return unlimited number of rows in CSV format and are cancelled after 1800 seconds.
Browser queries return maximum of 1000 rows in HTML format and are cancelled after 90 seconds.

```
SELECT
  VmaxRadius as r_max, -- The two variables we
  Vmax as v_max       -- want to extract
FROM
  RefL0100N1504_SubHalo -- The simulation
WHERE
  SnapNum = 28         -- The snapshot
```

1) Query area

2) Execute query

Query (stream)

Query (browser)

Help

Maximum number of rows to return to the query form: 10 ← 3)

Previous queries:

List of all queries executed so far in this session. Selecting a query will make it appear in the query window. The button will show all of them in a separate window. Refreshing that window will load the latest queries again.

SELECT VmaxRadius as r_max, -- The two variables we Vmax as v_max | Show All ← 4)

Demo queries: click a button and the query will show in the query window.

Holding the mouse over the button will give a short explanation of the goal of the query. These queries are also available on [this page](#).

Subhalo: SUB1 SUB2
Sizes: SIZE1
Magnitudes: MAG1

5) Demo queries

Metadata queries: The SQL statements under these buttons provide examples for querying and managing the state of a private database. Holding the mouse over the button will give a short explanation of the goal of the statement.

Available datasets:

- Simulations:

- Ref L0025N376

Simulation Model

Box length in comoving megaparsecs

Cube root of the initial number of particles per species

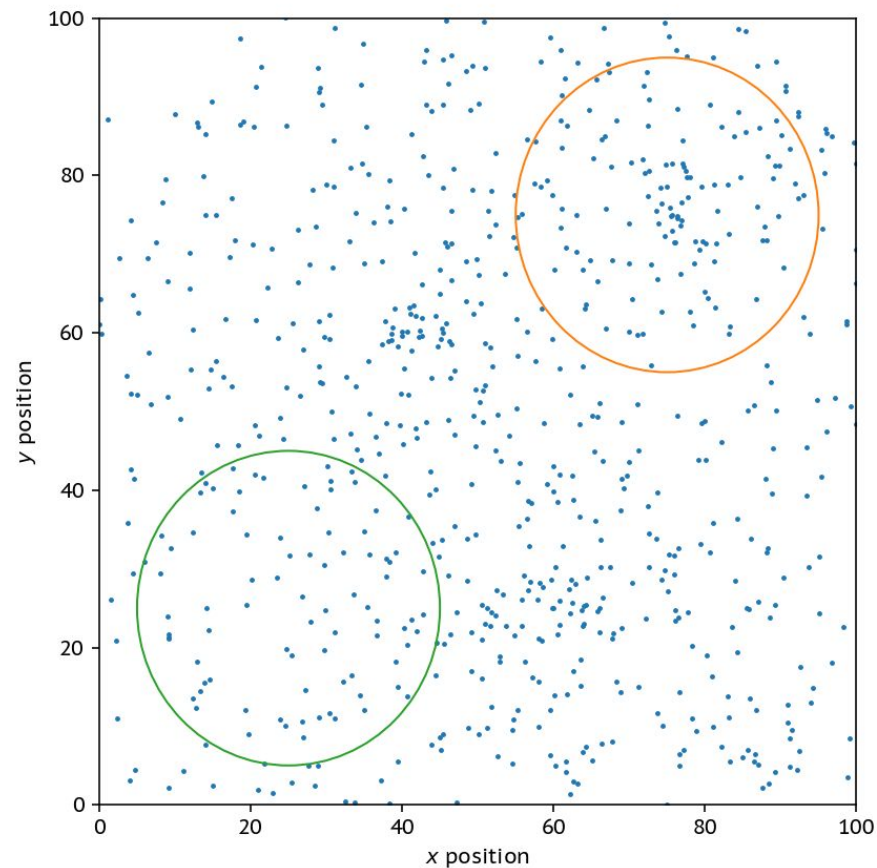
- Datasets:

- **Subhalo** - Main galaxy properties
- **FOF** - Halo properties
- **Sizes** - Galaxy sizes
- **Aperture** - Galaxy properties in 3D apertures
- **Magnitudes** - Galaxy photometry in the GAMA bands
- **Particle data**

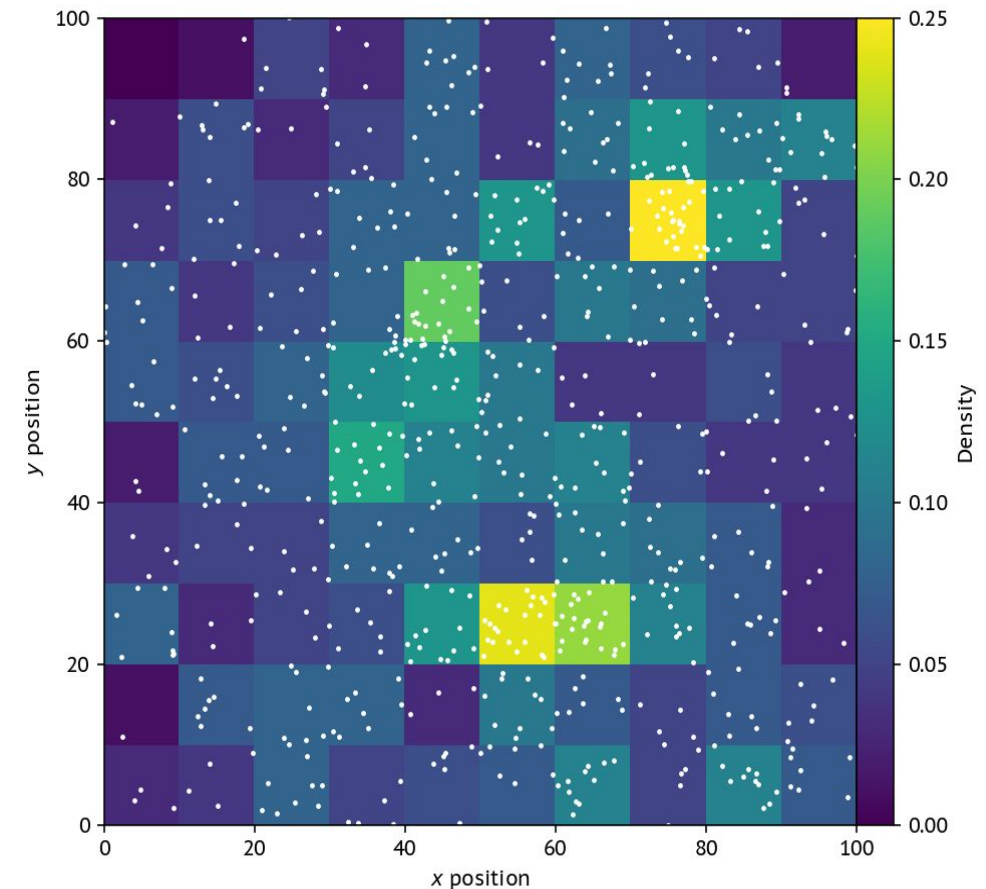
Example - 1

Visualisation - What is a smooth density field?

The problem:



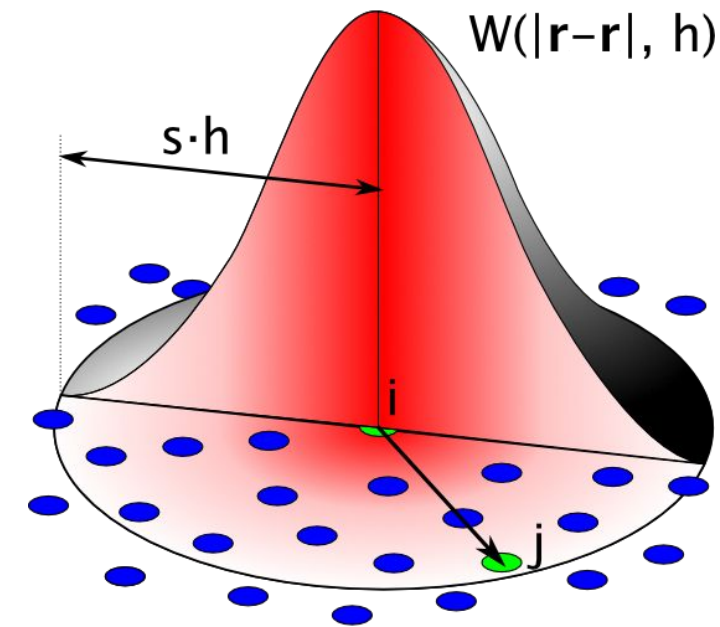
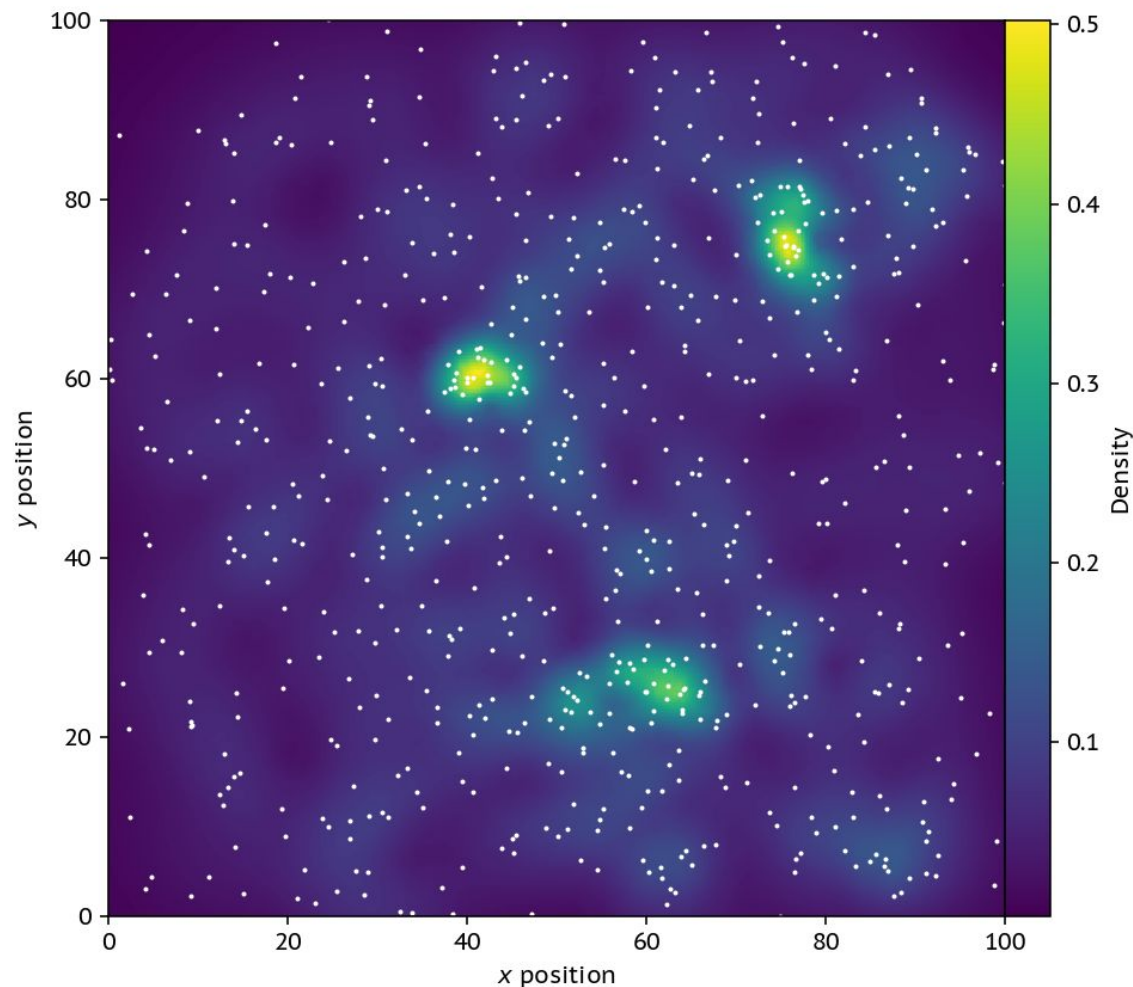
Solution 1: Build a grid



Example taken from: https://community.dur.ac.uk/joshua.borrow/blog/posts/density_from_particles/

Visualisation - What is a smooth density field?

Solution 2: Use a smoothed density estimate



Source: Wikipedia

The density at each point calculated using a kernel keeping the number of (weighted-)particles in each cell constant.

This kernel ensures that particles that are further away matter less to the density estimate.

Py-SPHViewer

Under the SPH (Smoothed-particle hydrodynamics) approximation, the projected density field at any pixel is the contribution of all particles within the “projected” kernel.

$$\rho(x) = \sum_j m_j W(|x - x_j|, h_j)$$

$$\Sigma(x, y) = \int \rho(x, y, z') dz' = \sum_j m_j \int W(|x - x_j|, h_j) dz'$$

$$\Sigma(x, y) = \sum_j m_j \tilde{W}(R_j, h_j)$$



You can download Py-SPHViewer here:

<https://github.com/alejandrobl/py-sphviewer>

Example - 2

Some additional examples using SPH visualisation:

- <https://www.youtube.com/watch?v=S7WCR39brw0>
- https://www.youtube.com/watch?v=2Fp_VK-xuKE

Questions?

Milky Way and Gaia

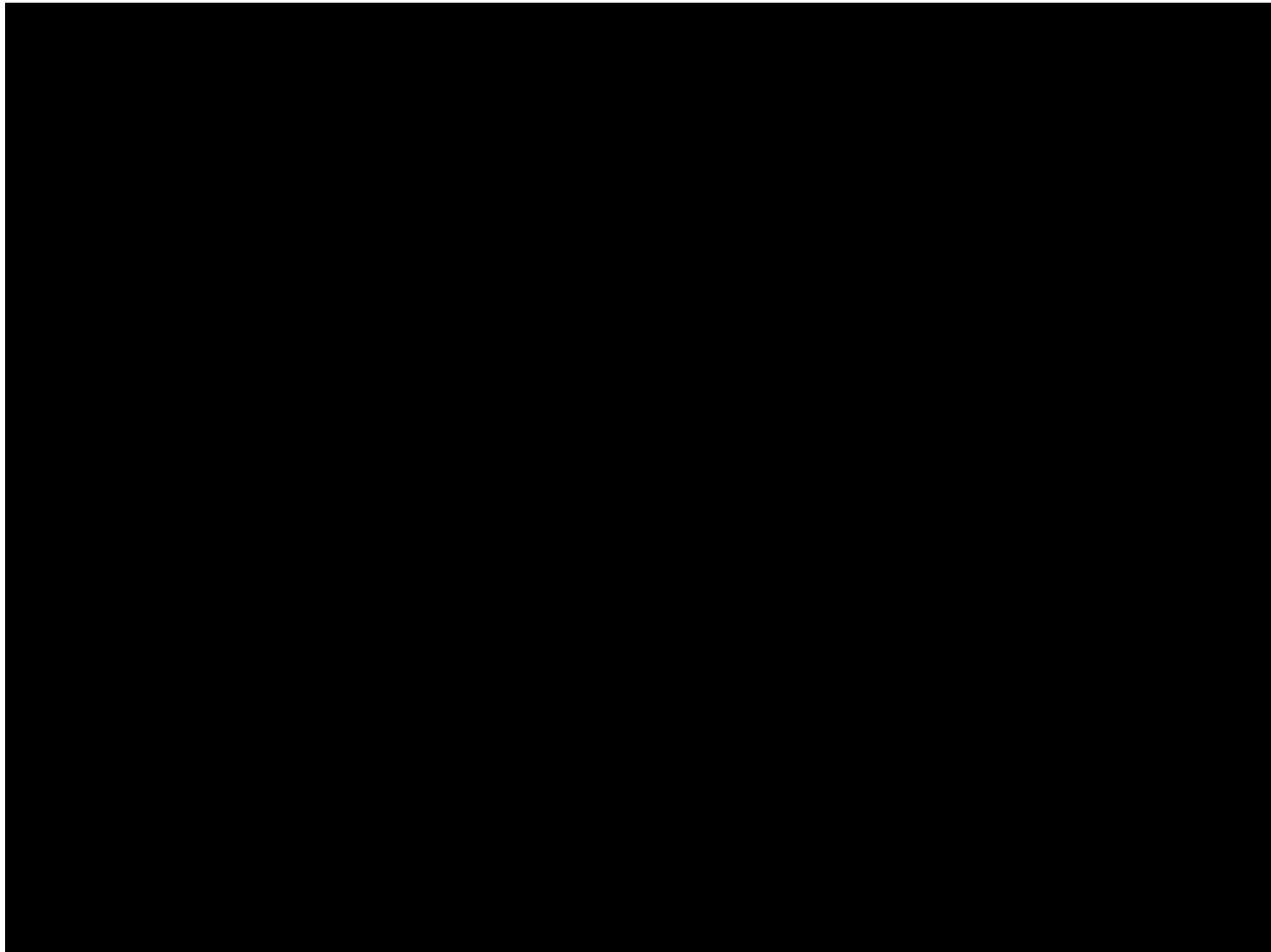
Andreea Font

How did the Milky Way form?

STARS

GAS

**ARTEMIS
simulations**





**Gaia is providing us
the first 3D map of the
Milky Way**



UNIVERSITY OF
LIVERPOOL

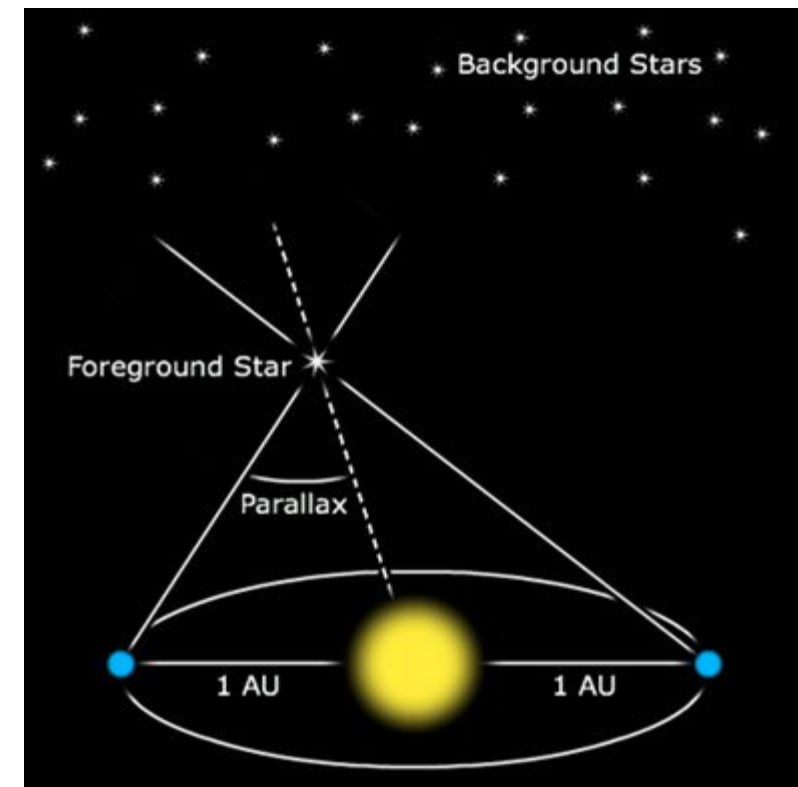


LIVERPOOL
JOHN MOORES
UNIVERSITY

UK
RI



Science and
Technology
Facilities Council



- Gaia can infer distances by measuring **parallax**
- Gaia DR2 contains parallaxes for **1.3 billion stars** with errors as small as 50 micro-arcseconds, equivalent to seeing a human hair from a distance of 500 km.
- Gaia is also taking precise measurements of the **brightness** (down to mag $G=20$), **colours** and **proper motions** of ~ 1 billion stars.

360° view of Gaia's sky

<https://www.youtube.com/watch?v=8cXURHmtf3I>

360° Parallax and proper motion on the sky

<https://www.youtube.com/watch?v=KyQdK56Qee0>

Gaia Archive:

<https://gea.esac.esa.int/archive/>

Login
(optional)

gaia archive

HOME SEARCH STATISTICS VISUALISATION HELP

Welcome to the Gaia Archive at ESA

Gaia is a European space mission providing astrometry, photometry, and spectroscopy of more than 1000 million stars in the Milky Way. Also data for significant samples of extragalactic and Solar system objects is made available. The Gaia Archive contains deduced positions, parallaxes, proper motions, radial velocities, and brightnesses. Complementary information on multiplicity, photometric variability, and astrophysical parameters is provided for a large fraction of sources.

Top Features

- Citation**
How to cite and acknowledge Gaia.
- Search**
Query for Gaia sources using an ADQL (Astronomical Data Query Language) interface in an asynchronous mode (UWS).
- Download**
Direct download of Gaia data files.
- Help**
For questions, suggestions or problem reports, contact the Helpdesk.
- Gaia Mission**
News, information, and resources on the Gaia mission for the scientific community.
- Statistics**
Show statistics of Gaia tables.
- Partners**
Partner data centres also serving Gaia data.

COPYRIGHT 2018 © EUROPEAN SPACE AGENCY. ALL RIGHTS RESERVED. [v2.9.1]

gaia archive

HOME SEARCH STATISTICS VISUALISATION HELP

Basic Advanced (ADQL) Query Results

Position File

Name
 Equatorial

Target in Circle Box

Name for

Radius

m54 resolved.

Search in:

▶ Extra conditions

▶ Display columns

Max. number of results:

Reset Form

Show Query

Submit Query

Job name:

Query examples

```

1 SELECT TOP 500
  gaia_source.source_id,gaia_source.ra,gaia_source.ra_error,gaia_source.dec,gaia_source.dec_error,gaia_source.parallax,gaia_source.parallax_error,gaia_sou
rce.phot_g_mean_mag,gaia_source.bp_rp,gaia_source.radial_velocity,gaia_source.radial_velocity_error,gaia_source.phot_variable_flag,gaia_source.teff_val,
gaia_source.a_g_val
2 FROM gaiadr2.gaia_source
3 WHERE
4 CONTAINS (
5   POINT('ICRS',gaiadr2.gaia_source.ra,gaiadr2.gaia_source.dec),
6   CIRCLE(

```

Ctrl+Space for query autocompletion

Reset Form

Submit Query

Astronomical Data Query Language (ADQL)

- an **SQL** `dialect` developed for astronomy by the Virtual Observatory.

Example 1:

Retrieve the source identifier ('source_id'), right ascension ('RA'), declination ('DEC'), and parallax ('parallax') for all sources in *Gaia DR2* that have parallaxes in the range 15–50 mas as well as G-band magnitudes ('photo_g_mean_mag') in the range 9–9.5 mag.

Syntax:

```
SELECT source_id, ra, dec, parallax  
FROM gaiadr2.gaia_source  
WHERE parallax between 15 AND 50 AND phot_g_mean_mag >= 9 AND  
phot_g_mean_mag <= 9.5
```


More on **SQL**:

<https://www.sqlcourse.com/>

ADQL Cookbook:

<https://www.gaia.ac.uk/data/gaia-data-release-1/adql-cookbook>

More technical document on ADQL:

<https://www.ivoa.net/documents/REC/ADQL/ADQL-20081030.pdf>

Gaia archive tutorials:

<https://gea.esac.esa.int/archive-help/>

Gaia DR2 Model (what parameters? units):

https://gea.esac.esa.int/archive/documentation/GDR2/Gaia_archive/chap_datamodel/

The result of the query can be downloaded in various formats:

- VOTable; CSV; FITS; JSON

Basic Advanced (ADQL) Query Results

16024424645490 X

source_id	ra deg	ra_error mas	dec deg	dec_error mas	parallax mas	parallax_error mas	phot_g_mean_mag mag	bp_rp mag	radial_velocity km.s ⁻¹
567304539159590528	1.943847263933078	2.2183004778840067	83.12859840421683	2.756514142970163	3.00000020417362	3.475613783408368	20.939615	0.5976505	
3414837379320591872	80.02290762770141	1.9465328828698256	22.88710089519733	1.4243181489062613	3.000000596066667	2.2737769575597175	20.760464	1.3391247	
5822151567899459968	239.0472862601741	0.955014342022537	-66.98889682574703	0.8227704830378849	3.00000065216071	1.5035975446501049	20.914507	1.7310066	
4167250675314428416	259.5514160201236	1.1641967677427485	-9.912790188719985	0.9086419255686005	3.0000001313622646	1.2489709348257345	20.667692	1.3010616	
141500278843232768	42.95468605883477	1.4715598278428972	37.20253861954247	1.1856796327153942	3.000000152252159	1.7110310689387886	20.696165	2.5031128	
2039147637754335488	288.9754089690946	0.12496466266918414	29.475322703391143	0.14187644941915395	3.0000001579249047	0.1655308649585397	16.90919	1.9975939	
1619244074077007616	226.35541022706713	0.9312738033931112	62.00466670054089	0.9908709390600131	3.0000002483662227	0.781459394239726	20.544878	1.5413342	
6254938822251371648	233.46202255275068	0.664707727662377	-19.34378285293941	0.5168971496988178	3.000000260012579	0.6809436523811101	19.646704	2.7502308	
5836729850822906240	185.1059865652506	0.24444911409000072	-77.35478669798637	0.22117033795155253	3.0000002769076595	0.27423828129253003	19.256058	3.1434574	
6038615548788017024	242.60299072044393	1.1012034920836296	-29.559970816174804	0.4631838379259978	3.0000003262728345	1.2284870904687217	20.519066	1.5972118	
4623106199028573568	88.19212843767284	0.653731725808769	-80.12776382114767	0.7922502370482454	3.0000003647225246	0.7197339206976358	20.423838	2.1796303	
4268279610528053760	288.293430501706	0.8948793024185012	3.3943705024953252	0.8872743050785302	3.000000373509193	0.6252419963135777	19.723185	1.6490498	
5990805897006793216	244.25372844199248	0.8809506644628131	-44.33483749717577	0.5302076879306866	3.0000004429439953	0.9009132517389731	19.997984	2.8739681	
4127732784311532544	256.9437022904397	1.4167252289152663	-20.83203845590637	1.1499734099674228	3.0000005275129786	1.7797986741500376	20.628622	1.3486176	
5933653729259316864	247.6997330473052	0.08721423579163519	-53.193412492148944	0.06554892875593755	3.0000006028003705	0.09870426143643522	17.190062	2.2877522	
3301620289051645952	59.26762490803188	0.036848956380880225	9.025956428245957	0.02064047241124432	3.0000006149079264	0.04247739914147331	11.725768	0.87062454	
6097862762908102656	213.41959512739965	1.042407430994955	-42.14846414905339	0.9146906229945896	3.000000630580003	1.3755530115362051	20.021347	1.8532658	

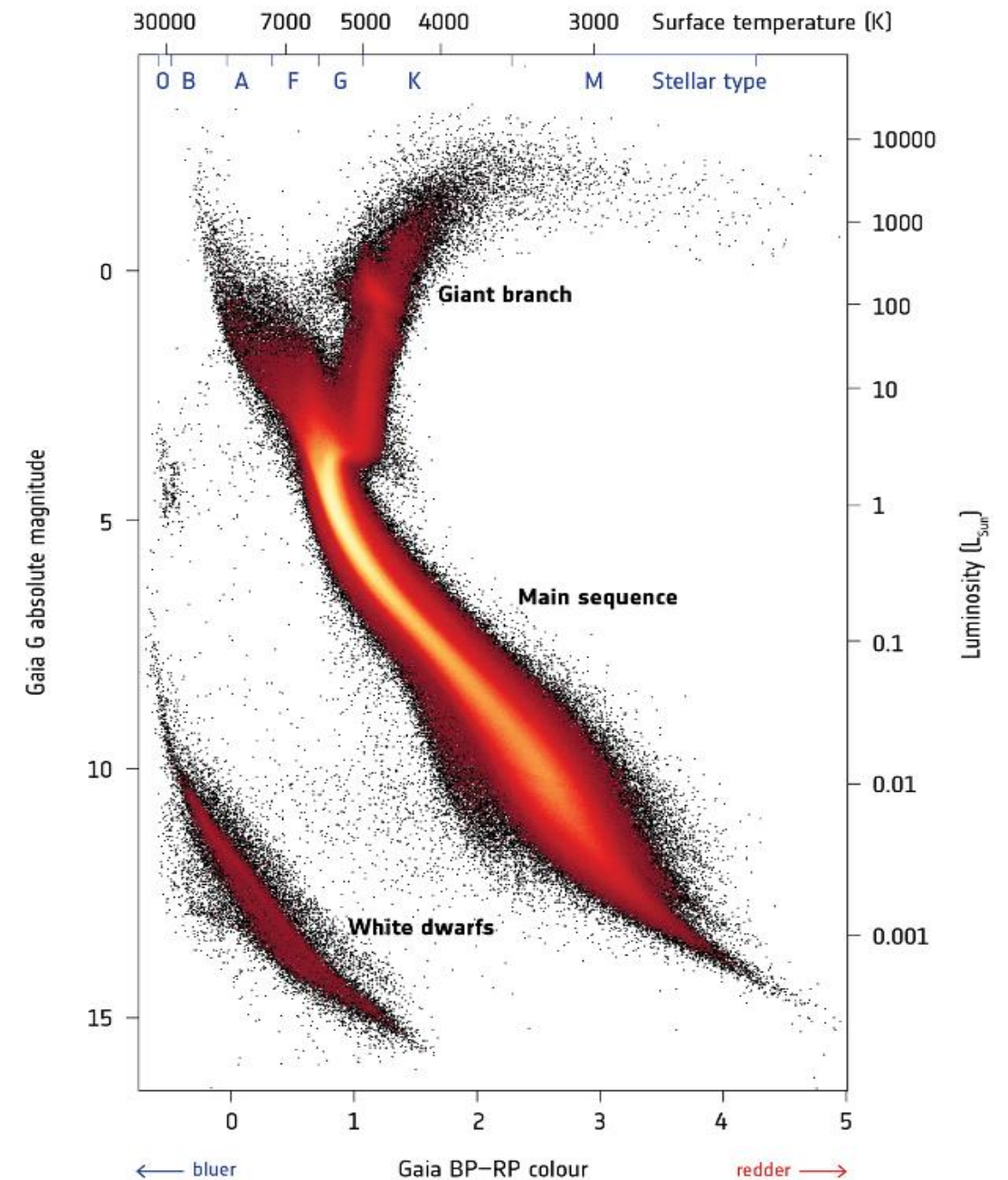
1-20 of 2,000 Gaia DR2 Data Model Show query in ADQL form CSV Download results

Download your data

→ GAIA'S HERTZSPRUNG-RUSSELL DIAGRAM

Hertzsprung - Russell diagram with Gaia

<https://www.youtube.com/watch?v=jutw-IOwriw>



Example 2:

Select for top 10000 stars the G-band magnitudes and BP-RP colours.

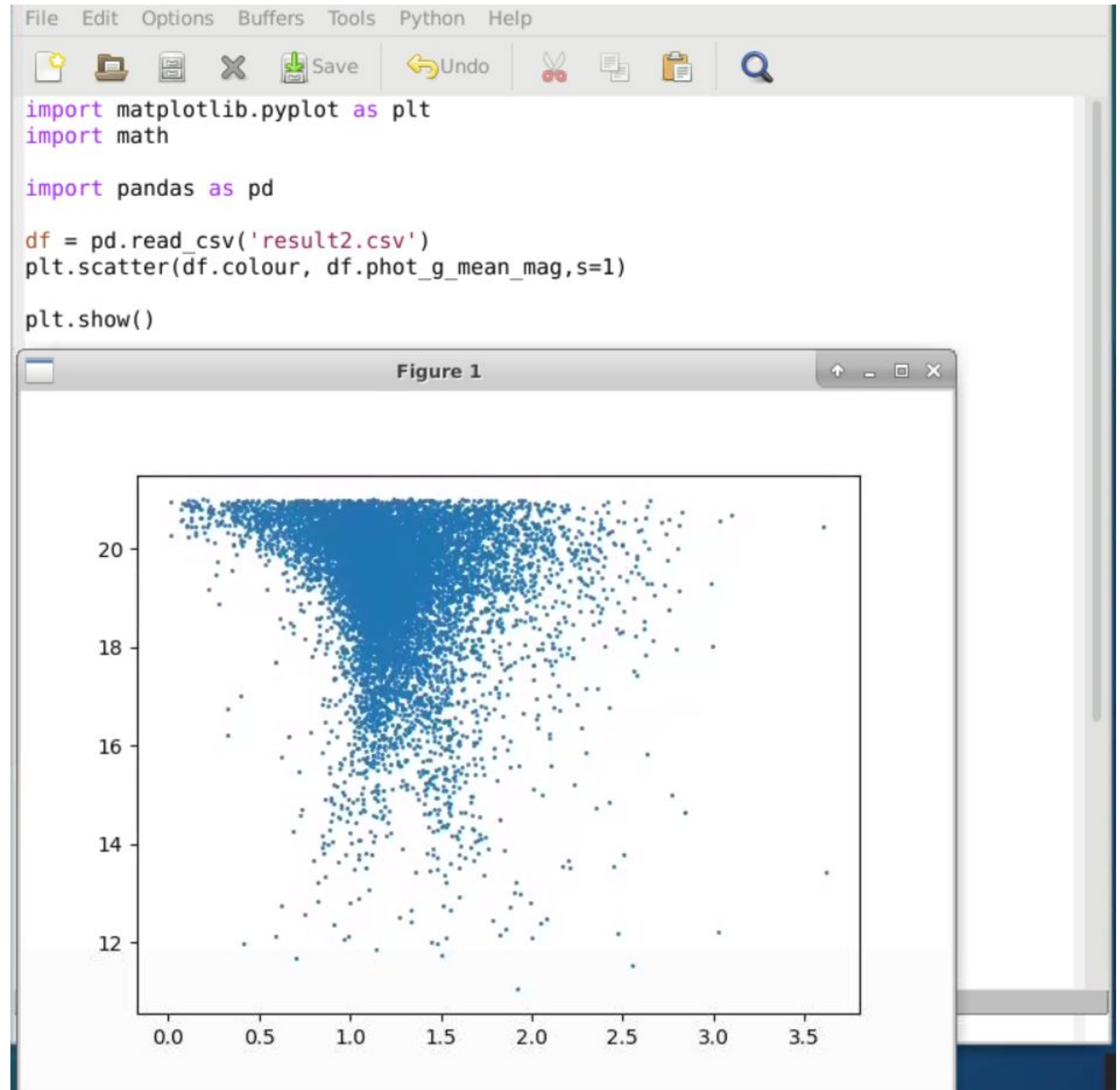
where BP and RP are 'blue' and 'red' pass bands specific to Gaia.

ADQL query:

```
1 SELECT TOP 10000 source_id, phot_g_mean_mag, (phot_bp_mean_mag-phot_rp_mean_mag) as colour
2 FROM gaiadr2.gaia_source
3 WHERE (phot_bp_mean_mag-phot_rp_mean_mag) > 0
4
```


With the result saved in a file, one can plot a colour - magnitude diagram.

Here is an example of Python code for data in csv format:



Example 3:

Select top 10000 stars in the ‘Solar neighborhood’, i.e. a spherical region of radius of 300 parsec (1 pc = 3.26 lyr),

or, equivalently, with parallax > 3

ADQL query

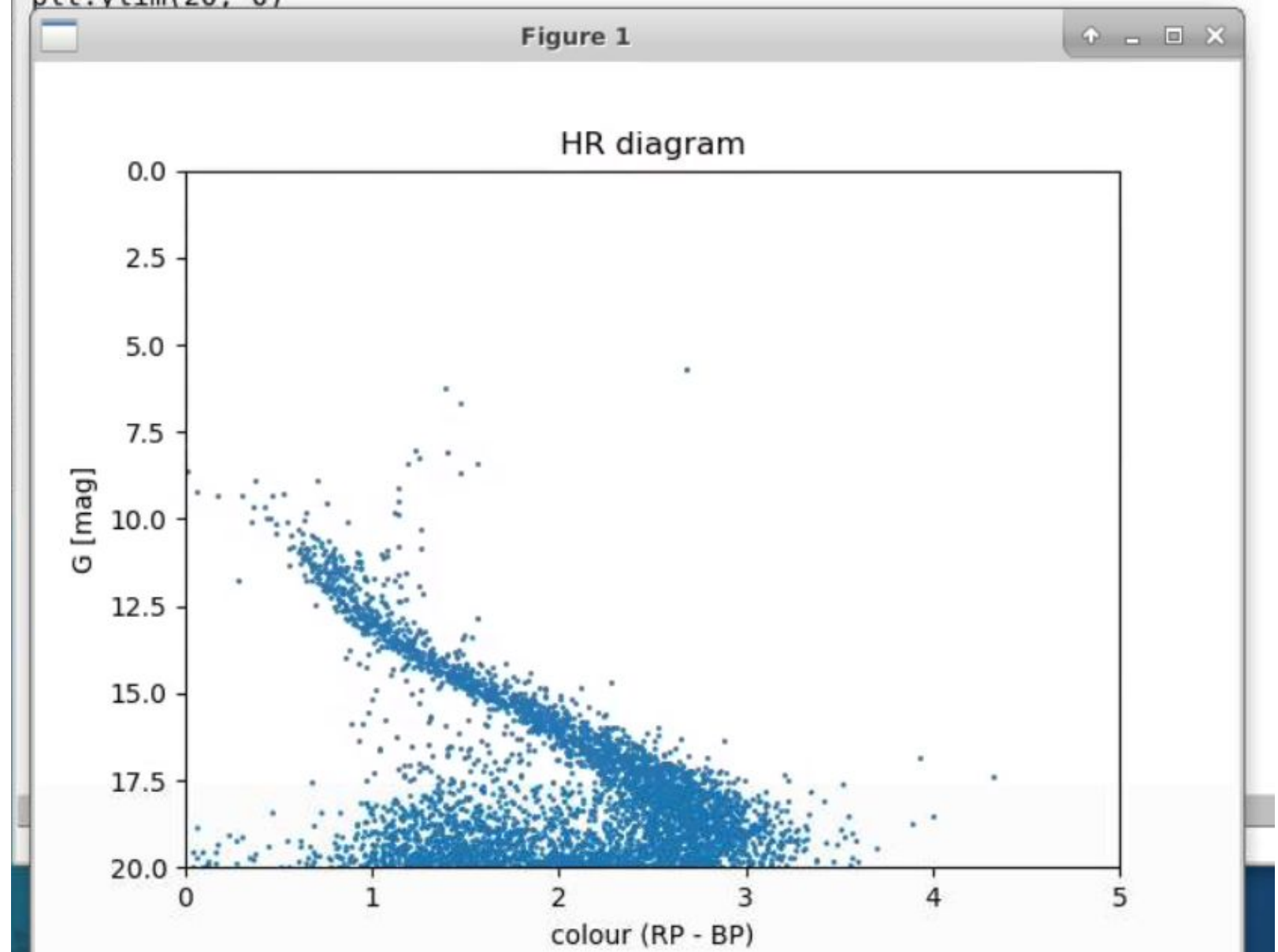
(note: here I’ve limited the search only to 10000 objects. The real region contains millions of stars; to get them all, omit ‘TOP’; the query will take longer)

```
1 SELECT TOP 10000
   gaia_source.source_id,gaia_source.ra,gaia_source.ra_error,gaia_source.dec,gaia_source.dec_error,gaia_source.parallax,gaia_source.parallax_error,gaia_source.phot_g_mean_
   mag,gaia_source.bp_rp,gaia_source.radial_velocity,gaia_source.radial_velocity_error,gaia_source.phot_variable_flag,gaia_source.teff_val,gaia_source.a_g_val
2 FROM gaiadr2.gaia_source
3 WHERE(parallax > 3)
4 |
```


.. and the colour -
magnitude diagram
from this query:

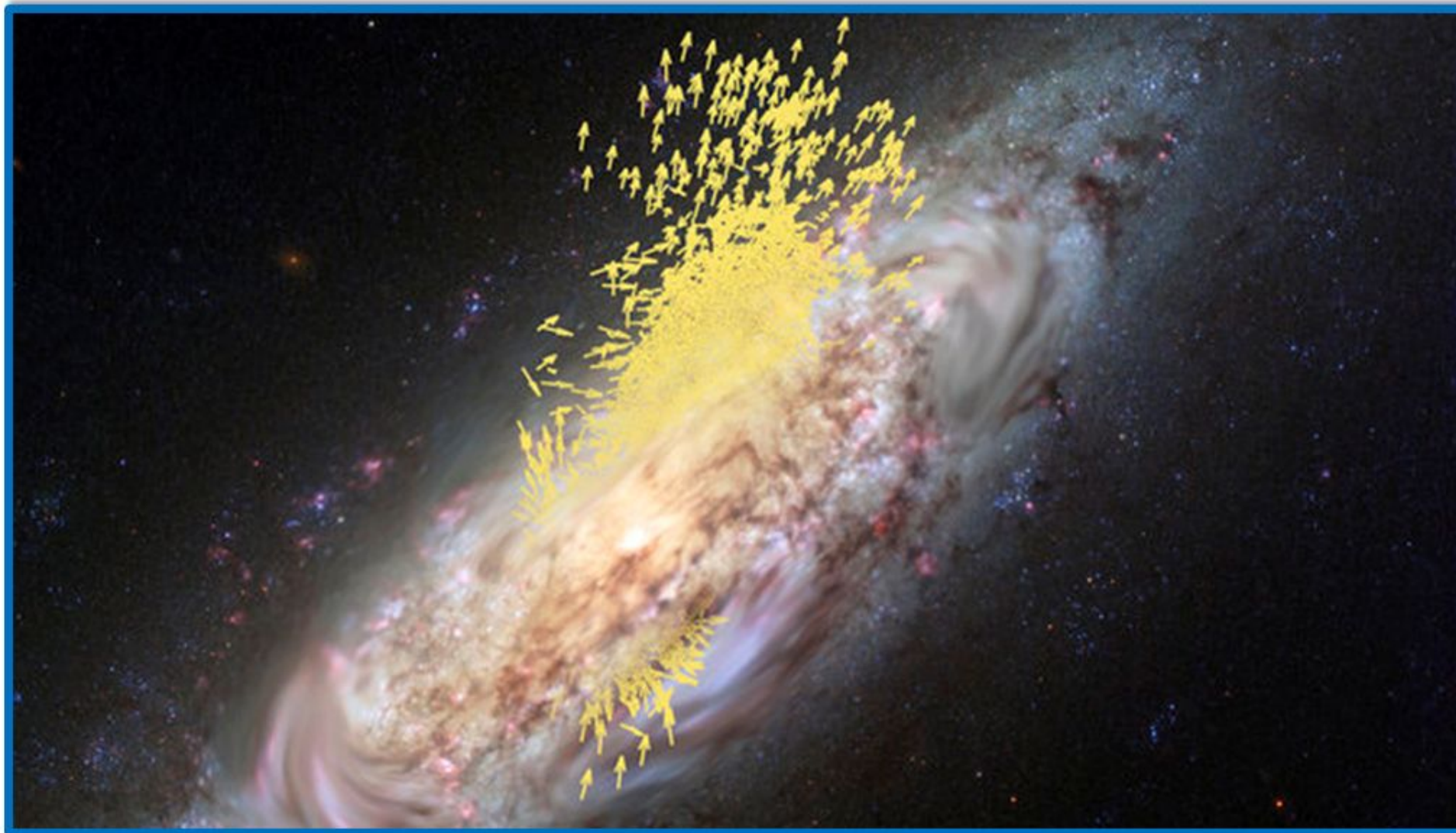
```
import matplotlib.pyplot as plt
import math
import pandas as pd

df = pd.read_csv('result-SN.csv')
plt.scatter(df.bp_rp,df.phot_g_mean_mag,s=1)
plt.title('HR diagram')
plt.xlabel('colour (RP - BP)')
plt.ylabel('G [mag]')
plt.xlim(0, 5)
plt.ylim(20, 0)
```



Major discovery: “*Gaia Enceladus*”

- *an ancient merger of the Milky Way with another galaxy the size of the LMC*



UNIVERSITY OF
LIVERPOOL



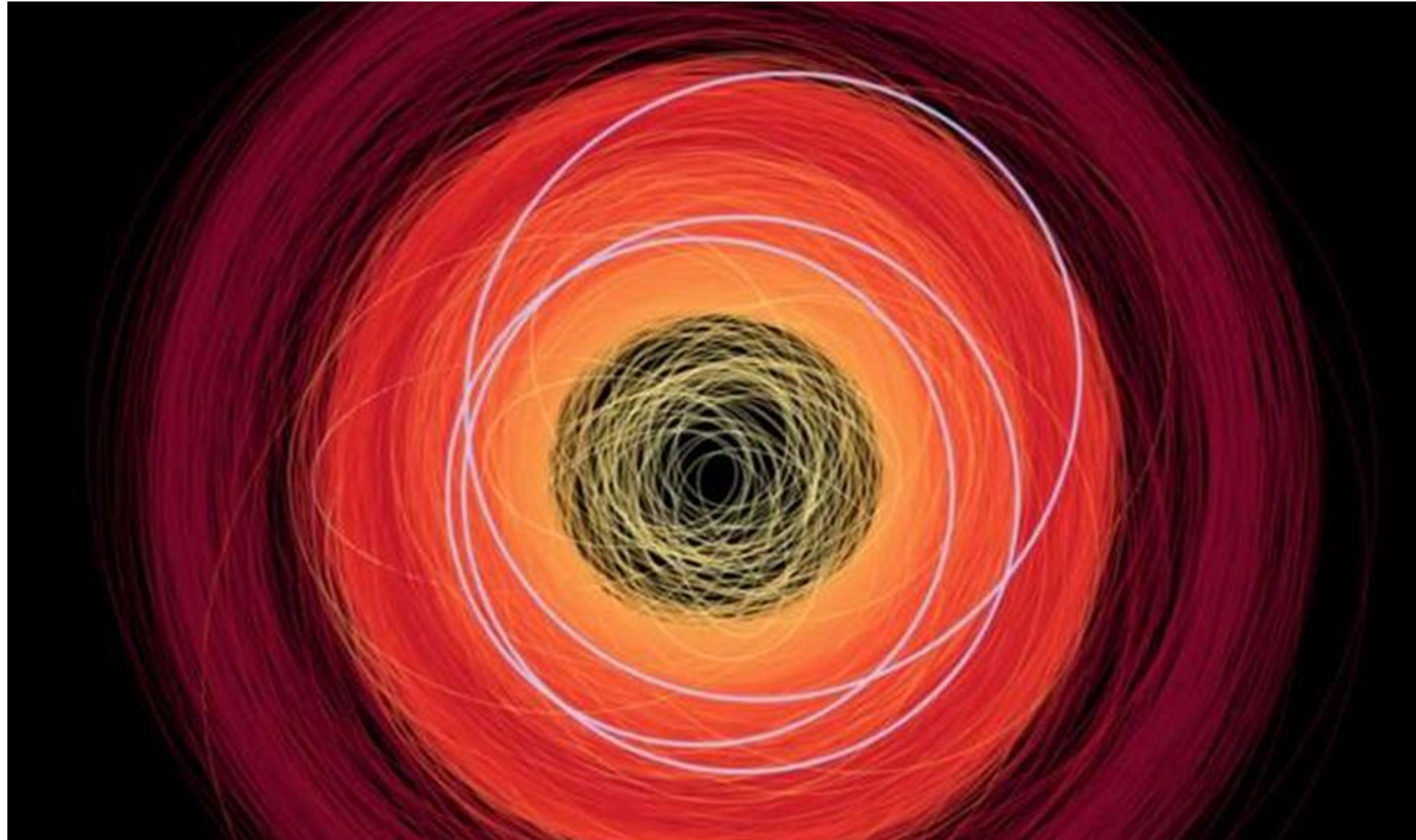
LIVERPOOL
JOHN MOORES
UNIVERSITY

UK
RI



Science and
Technology
Facilities Council

Orbits of ~14,000 known asteroids with Gaia



Grey shows asteroids discovered by Gaia.

Asteroids shown in bright red and orange hues are **main-belt asteroids**, located between the orbits of Mars and Jupiter.

Trojan asteroids, found around the orbit of Jupiter, are shown in dark red. In yellow are the orbits of several tens of **near-Earth asteroids** (<1.3 AU).



UNIVERSITY OF
LIVERPOOL



LIVERPOOL
JOHN MOORES
UNIVERSITY

UK
RI



Science and
Technology
Facilities Council

Next Gaia release (Gaia DR3) : 3 Dec 2020

(mission will continue until 2022 and maybe beyond)

It is estimated that in the next few years Gaia will discover:

- ✓ hundreds of thousands of asteroids and comets within our Solar System
- ✓ seven thousand planets beyond our Solar System (exoplanets)
- ✓ tens of thousands of 'failed' stars, called brown dwarfs
- ✓ twenty thousand exploding stars, called supernovae
- ✓ hundreds of thousands of distant active galaxies, called quasars.

Gaia will also give us:

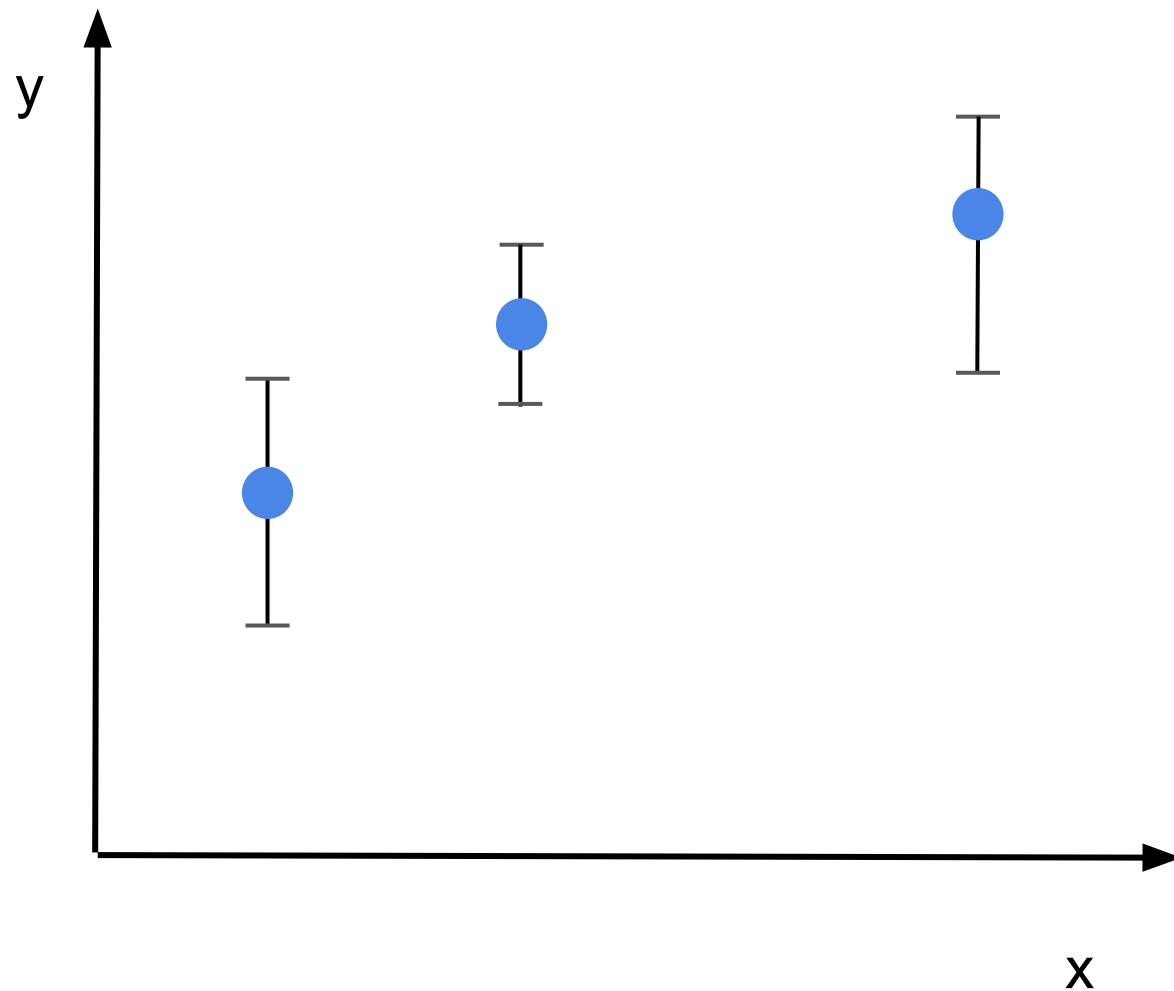
- ✓ a refined measurement of the cosmic distance scale
- ✓ new tests of general relativity

Questions?

Parameter Inference & MCMC

Juliana Kwan

Suppose you needed to fit a straight line to some data...



My model is: $y = mx + b$

$$\chi^2 = \sum_i \frac{[y_i - \tilde{y}(x_i | m, b)]^2}{\sigma_i^2}$$

Work out the best fitting values of m, b by minimising χ^2 .

Need to solve $\partial\chi^2/\partial\alpha_i = 0, \alpha_i = (m, b)$

The Bayesian approach:

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)}$$

Bayes' Theorem

$P(H|D)$ is the probability of the hypothesis given the data (**posterior**; this is what we want)

$P(D|H)$ is the probability of the data given the hypothesis (**likelihood**)

$P(H)$ is the **prior** hypothesis

$P(D)$ is the probability distribution of the data (important for model comparison)

Bayesian parameter estimation involves calculating the likelihood, $P(D|H)$:

$$\mathcal{L} \propto \exp \left[-\frac{1}{2} \sum_i (y_i - \tilde{y}(x_i)) C_{ij}^{-1} (y_j - \tilde{y}(x_j)) \right]$$

You'll notice that:

$$\log \mathcal{L} \propto -\frac{1}{2} \chi^2$$

This is true for Gaussian distributions only!

We must also choose a prior on the parameters, m , b , e.g:

$$\pi(m) = (m_{\max} - m_{\min})^{-1} \quad \pi(b) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{b-\mu}{\sigma}\right)^2\right] \quad \pi(b) = 1/b$$

This represents any external knowledge that we already have about the system, e.g. a weighted coin.

Now that we have the likelihood and prior, how should we calculate the posterior?

Markov Chain Monte Carlo Analyses

In real world applications, we often have many parameters, which makes evaluating the posterior expensive.

Fortunately there are numerical methods to deal with this.

The Metropolis-Hastings algorithm:

1. Choose a starting point, x_0 , calculate $L(x_0)$ (close to what you think the parameters should be)
2. Take a step to x_1 by calculating $L(x_1)$
3. If $L(x_1) \geq L(x_0)$, then keep x_1 by adding it to the chain
4. If $L(x_1) < L(x_0)$, generate a random number, $r \sim U(0,1)$
 - a. If $r \geq L(x_1)/L(x_0)$, keep x_1 by adding it to the chain
 - b. If $r < L(x_1)/L(x_0)$, discard x_1 (add x_0 to the chain)
5. Repeat steps 2 - 4 until convergence is reached.

Markov Chain Monte Carlo Analysis

At the end of the process, you will have a chain (list of parameter values) that represents the combination of the likelihood and the prior. The density of points in the chain is the posterior.

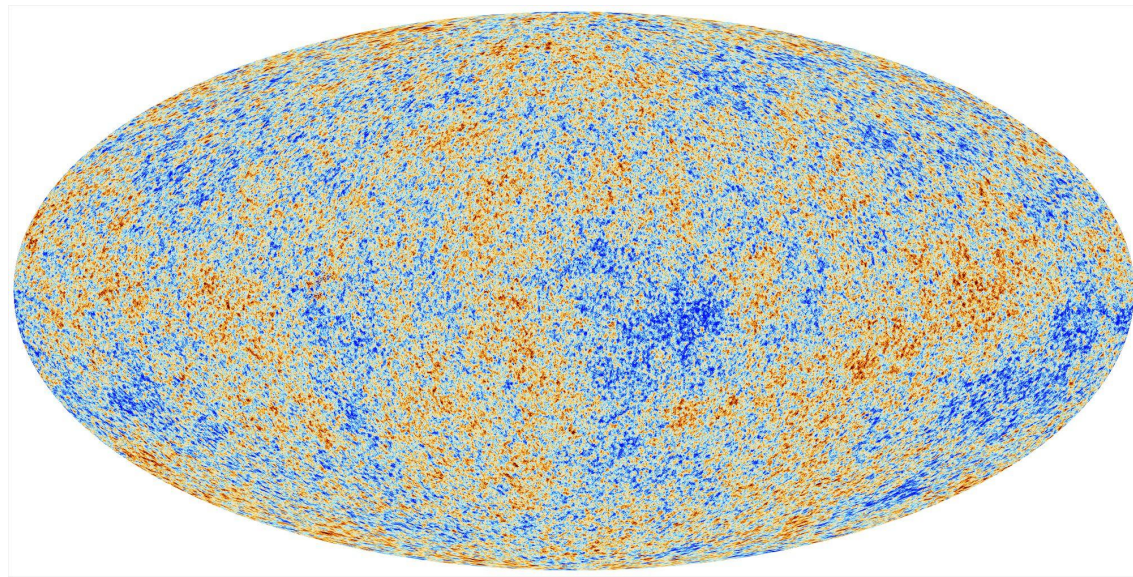
Some caveats:

- You must check for convergence. MCMC gives the correct posterior in the limit of infinite samples.
- Be careful in your choice of step size - too small and the chain will get stuck around the peaks of the posterior, too large and the chain can 'step over' the peaks.
- Multimodal posteriors require special treatment; see nested sampling

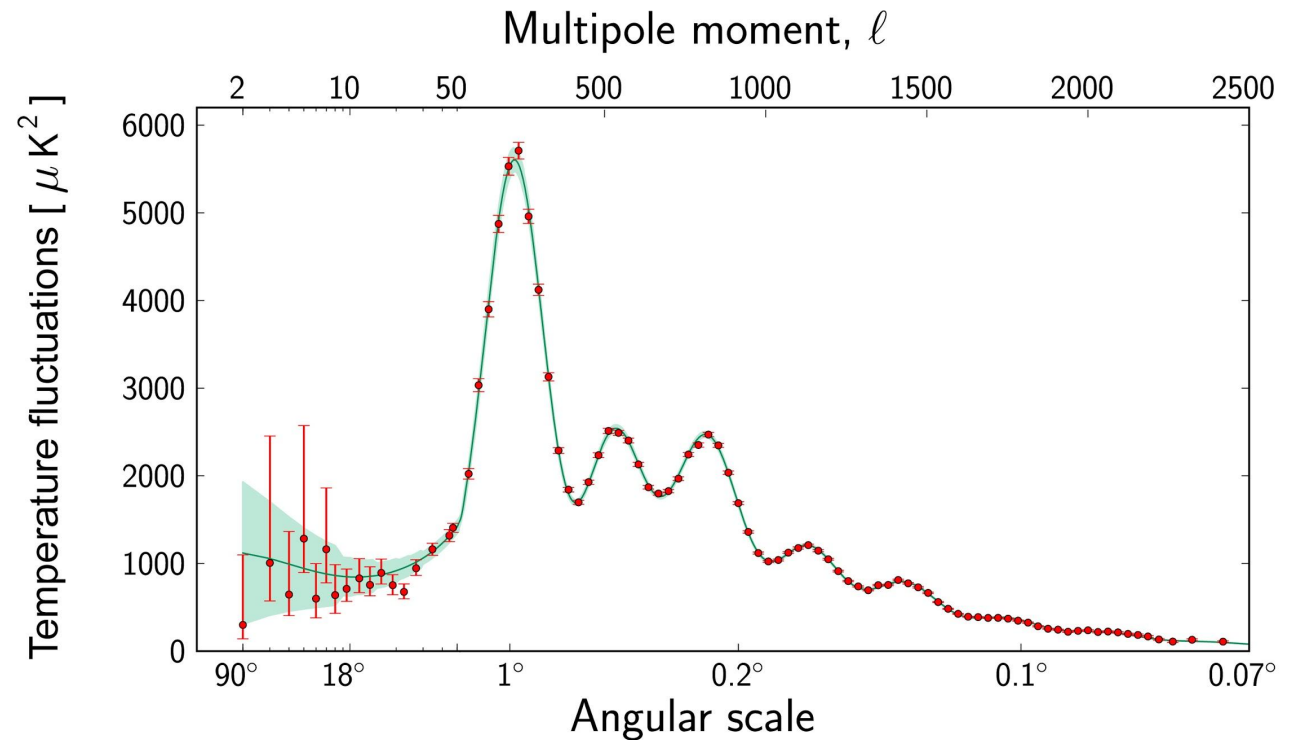
Helpful packages for parameter inference:

- CosmoMC and GetDist
 - <https://cosmologist.info/cosmomc/>
- CosmoSIS
 - <https://bitbucket.org/joezuntz/cosmosis/wiki/Home>
- emcee
 - <https://emcee.readthedocs.io/en/stable/>
- dynesty
 - <https://github.com/joshspeagle/dynesty>
- anesthetic
 - <https://anesthetic.readthedocs.io/en/latest/>

Example: Planck 2018 likelihood



Map of temperature fluctuations in the early universe



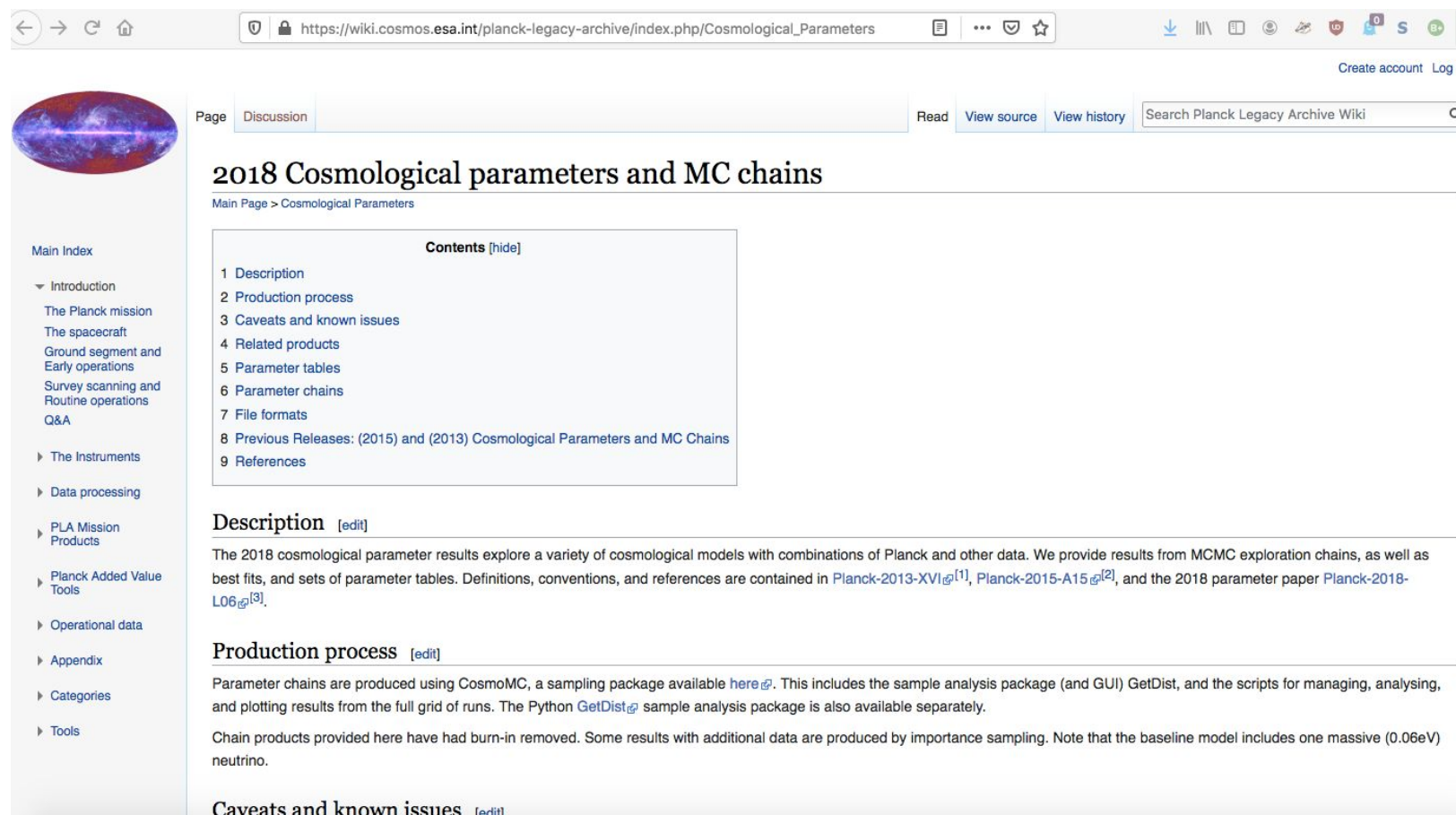
Spectrum of fluctuations as a function of angular size

Question: What model parameters best fit the observed fluctuations?

Example: Planck 2018 likelihood

Download the chains:

https://wiki.cosmos.esa.int/planckpla/index.php/Cosmological_Parameters



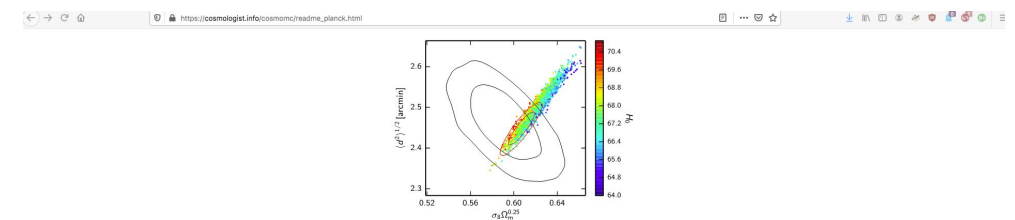
The screenshot shows a web browser displaying the Planck Legacy Archive Wiki page for '2018 Cosmological parameters and MC chains'. The browser's address bar shows the URL: https://wiki.cosmos.esa.int/planck-legacy-archive/index.php/Cosmological_Parameters. The page features a navigation menu on the left with categories like 'Main Index', 'Introduction', 'The Planck mission', 'The Instruments', 'Data processing', 'PLA Mission Products', 'Planck Added Value Tools', 'Operational data', 'Appendix', 'Categories', and 'Tools'. The main content area has a 'Page' tab selected, with sub-tabs for 'Discussion', 'Read', 'View source', and 'View history'. A search bar is located at the top right. The page title is '2018 Cosmological parameters and MC chains', with a breadcrumb trail 'Main Page > Cosmological Parameters'. A 'Contents' table of contents is displayed, listing sections from '1 Description' to '9 References'. The 'Description' section is expanded, providing details about the 2018 cosmological parameter results, including MCMC exploration chains, best fits, and parameter tables. It references 'Planck-2013-XVI', 'Planck-2015-A15', and 'Planck-2018-L06'. The 'Production process' section explains that parameter chains are produced using CosmoMC and GetDist. The 'Caveats and known issues' section is partially visible at the bottom.

If you want to run them for yourself (long!):

Download likelihood code: <https://pla.esac.esa.int/pla/#cosmology>

Install into CosmoMC: https://cosmologist.info/cosmomc/readme_planck.html

The screenshot shows the Planck Legacy Archive website. The header includes the European Space Agency logo and the text 'SCIENCE & TECHNOLOGY'. The main content area is titled 'Planck Legacy Archive' and features a navigation menu with categories: 'Cosmological parameters', 'CMB angular power spectra', 'Likelihood', 'Lensing products', and 'Noise covariance matrices'. The 'Likelihood' category is selected. Below the navigation, there is a 'RESULTS' section with a table of products. The table has columns for 'Description', 'File name', and 'Size'. The products listed include Planck 2018 TT, TE, EE, TTTEEE, TTEE and TTEE alternative high-ell CMBspec likelihoods (7.6 GB), Planck 2018 TE, EE binned high-ell Plik likelihood (5.1 GB), Planck 2018 LFI-based low-ell likelihood for TEB (180.7 MB), Planck 2018 data sets for baseline (57.5 MB), Planck 2018 TE and EE high-ell foreground- and nuisance-marginalized Plik_lite likelihoods (3.1 MB), Planck 2018 likelihood code (2.3 MB), and Planck 2018 extended ell range lensing likelihood (796.1 KB).



CosmoMC and Plotting with Planck Likelihood and Chains

The Planck likelihood code (PLClik) and parameter chains are available from the [Planck Legacy Archive](#). The Planck lensing and Birop-Kock Planck likelihoods are included with the cosmoc installation and do not need to be installed separately.

Using the Planck likelihood with CosmoMC

- Note you need to use Intel fortran (ifort) 14 or higher to build CosmoMC, or gfortran (gcc) 6 latest (easily available for testing in [virtual environments](#)) so make sure you have that configured before you start.
- Download and install the Planck likelihood code and baseline data files to somewhere convenient from [Planck Legacy Archive](#) (see also [description](#))
 - Extract and install the likelihood by doing:

```
tar xvf3 COM_Likelihood_Code-v3.0.tar.gz
cd plc-3.0
./waf configure --install_all_deps
(you may need to change the options on this line depending on your installation; --install_all_deps may not be needed; see the plc_2.0 readme.md for details)
./waf install
```
 - Run `./bin/click_profile.sh` and edit your `~/.bashrc` file (or equivalent) to include it in future sessions, or from command line:

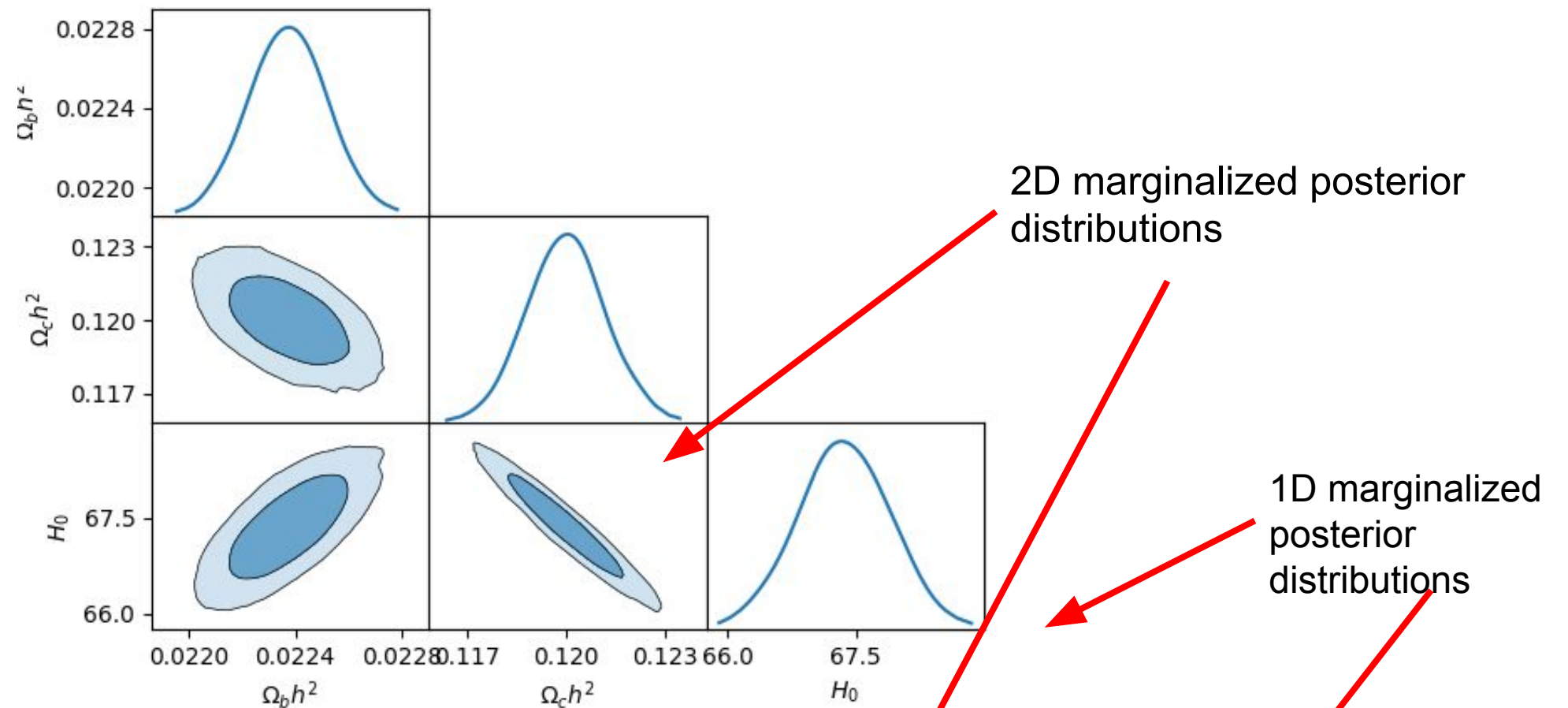
```
source ~/.bin/click_profile.sh
echo "export PATH=$PATH:$HOME/bin/click_profile.sh" >> ~/.bashrc
```

Here I assume you have installed CosmoMC in a directory called COSMOMC_PATH. If you haven't done it already, also add CosmoMC's python path (for plotting and analysis of chains) to your `~/.bashrc`:

```
export PYTHONPATH=$COSMOMC_PATH/python:$PYTHONPATH
```
 - Make sure you have also downloaded and extracted the Planck likelihood data files that you want (COM_Likelihood_Data-baseline...)
 - Then change to your COSMOMC_PATH root directory and make a symbolic link to your data file installation:

```
cd COSMOMC_PATH
ln -s $HOME/$PLANCK_ARCHIVE_PATH
```


pip install anesthetic (use flag --user if no root access)



```
import matplotlib.pyplot as plt
from anesthetic import MCMCSamples
mcmc_root =
'plikHM_TTTEE_lowl_lowE_lensing/base_plikHM_TTTEE_lowl_lowE_lensing'
mcmc=MCMCSamples(root=mcmc_root)
mcmc.plot_2d(['omegab2','omegach2','H0'], types={'lower':'kde','diagonal':'kde'})
```

Why does this work?

The file directory contains

- .paramnames file

This lists the names of the parameters and their latex labels

- .ranges file

This lists the min, max values of each parameter. Can be 'none'

If using CosmoMC, these files are automatically generated.

Some useful commands...

- `mcmc.mean()`
- `mcmc.median()`
- `mcmc.std()`
- `mcmc.describe()`: prints mean, std dev, confidence levels for each parameter, max/min values
- `mcmc.info()`: prints names of each column, data type, no. of entries.

Bayesian statistics is a very powerful tool for analysing large datasets.

Two functions:

- Parameter inference: given a certain model, what are the best fitting parameters to the observed data?
- Model selection: which model is better able to fit the data?

There are many publically available packages to help you!

Questions? Email: j.kwan@ljmu.ac.uk or message via Slack

Questions?