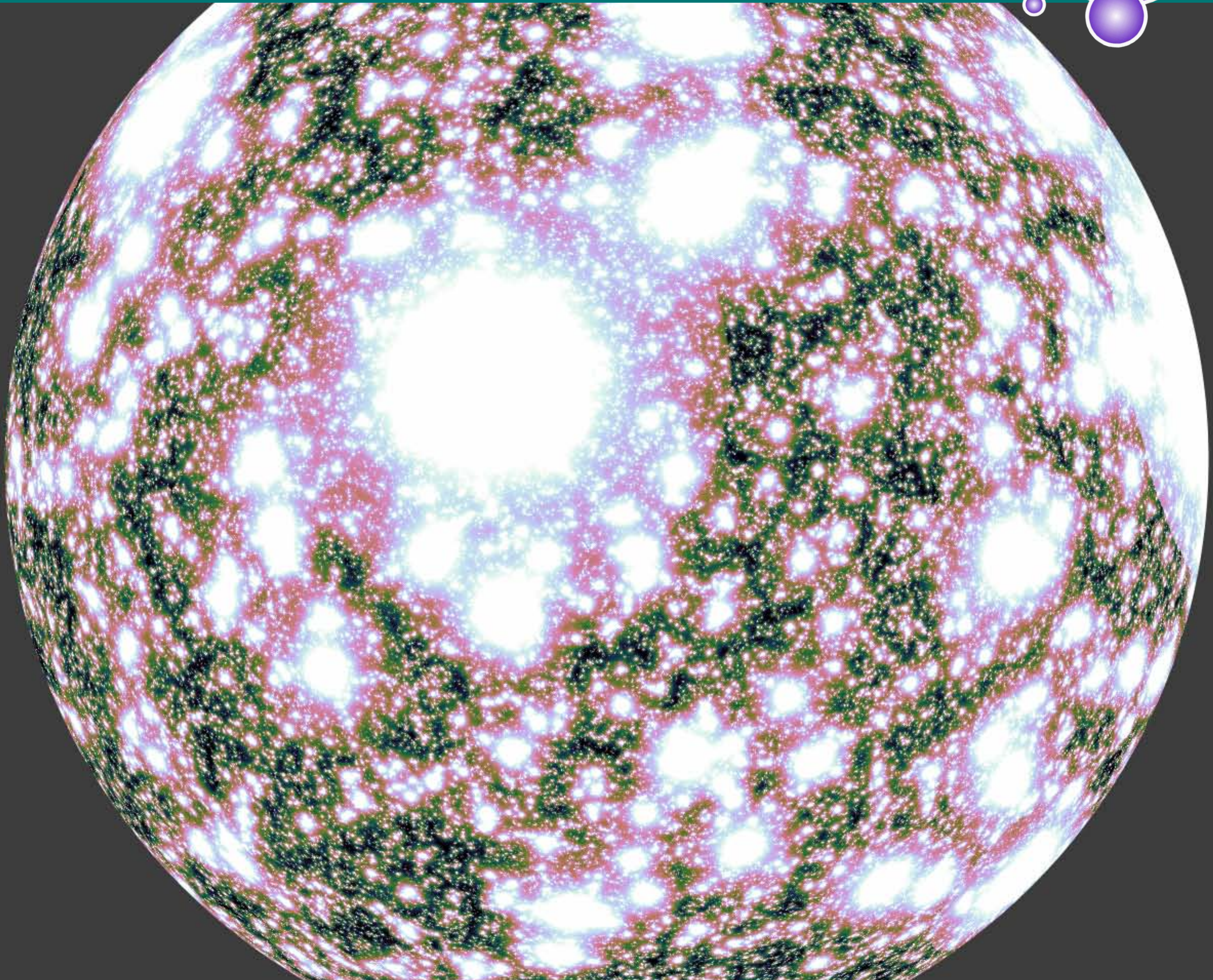
CLUMPY tutorial

Moritz Hütten, MPP Munich
Céline Combet, LPSC Grenoble

for the CLUMPY developers:
Vincent Bonnivard, Emmanuel Nezri, Aldée Charbonnier, David Maurin

ISAPP Dark Matter School ~~2020~~ 2021,
IFT UAM-CSIC Madrid, June 28 2021



MAX-PLANCK-INSTITUT FÜR PHYSIK

Enigmass
The enigma of mass

Who am I?

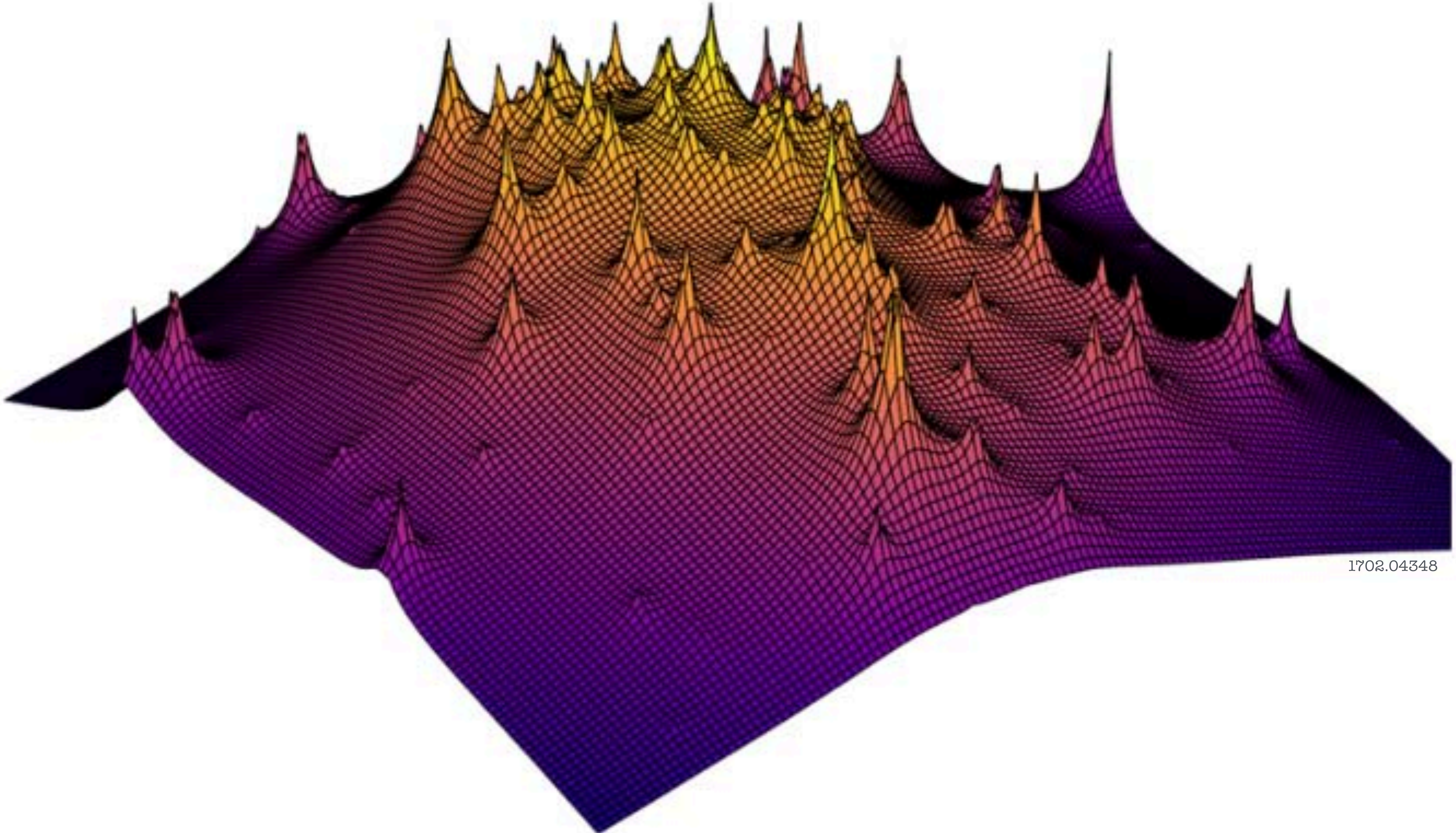
- Did my PhD in 2017 at DESY Zeuthen (close to Berlin, Germany) in VERITAS and CTA: contributed to Clumpy v2 and v3.
- One year Postdoc at Humboldt University Berlin
- Since 2018 Postdoc at the Max Planck Institute for Physics in Munich (Germany)
- Moving to ICRR Tokyo in two weeks

- Working on Indirect DM detection with MAGIC and data analysis of LST-1 telescope (together with MAGIC).

Program for the tutorial

- 20 Minutes Intro, Recap, and first steps into the code together with all of you
- **First exercise in small groups: 30 minutes**
- Solution of first exercise all together + Q&A: 15 minutes
- **Second exercise in small groups: 30 minutes**
- Solution of second exercise all together + Q&A: 15 minutes
- Close-out further questions: 10 minutes

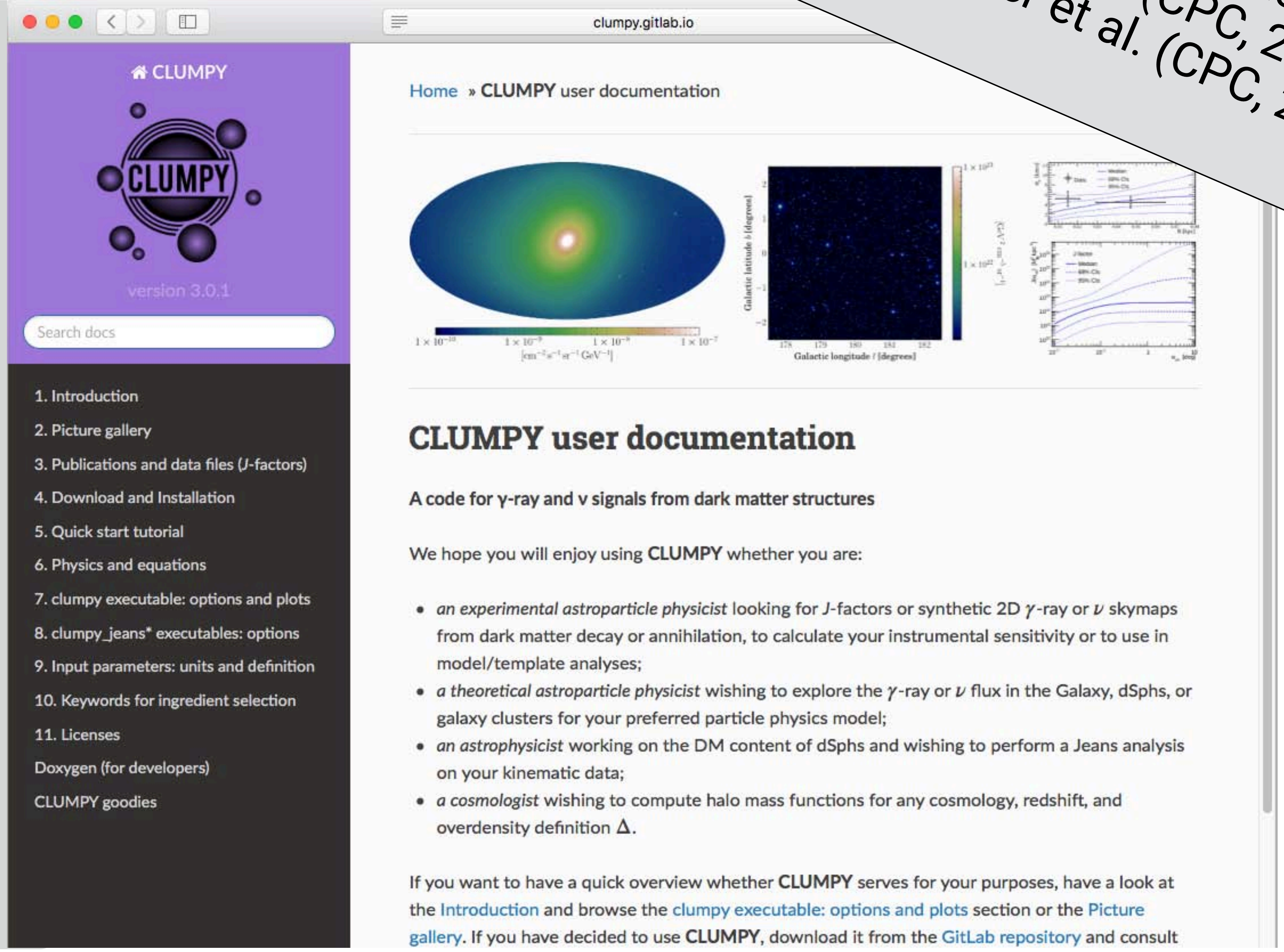
1. Introduction: What is CLUMPY for?



What is CLUMPY?

https://lpsc.in2p3.fr/clumpy/
Hütten et al. (CPC, 2018), arXiv:**1806.08639**
Bonnivard et al. (CPC, 2016), arXiv:**1506.07628**
Charbonnier et al. (CPC, 2012), arXiv:**1201.4728**

- Open-source code, written in C/C++
- Public development on GitLab
- Depends on:
  - gsl
  - Heasarc's cfitsio
  - ~~HEALPix~~ (shipped with the code)
  - CERN's ROOT (optional)
  - GreAT (lpsc.in2p3.fr/great, optional)
  - CLASS (optional)
- Runs on Linux and MacOS X
- Extensive web documentation



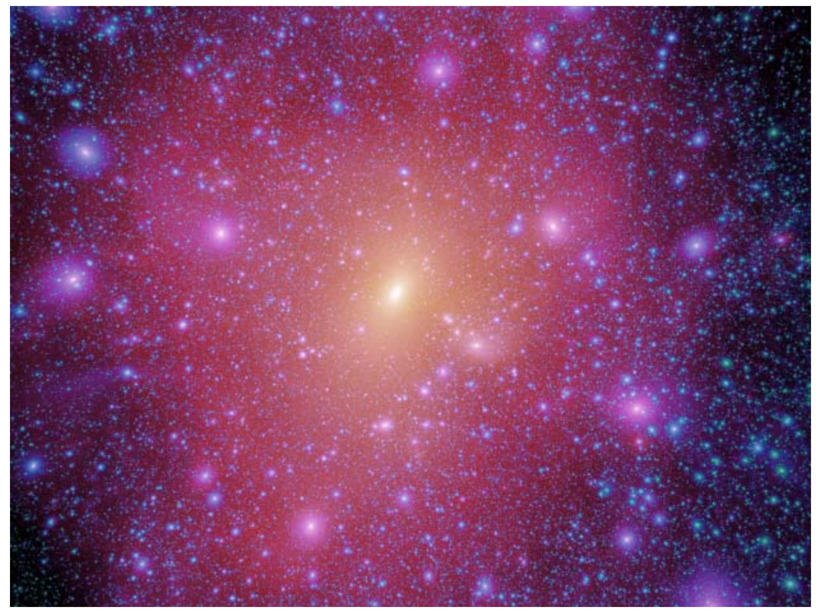
Provide the community reproducible models for J-factors and prompt γ-ray/ν fluxes

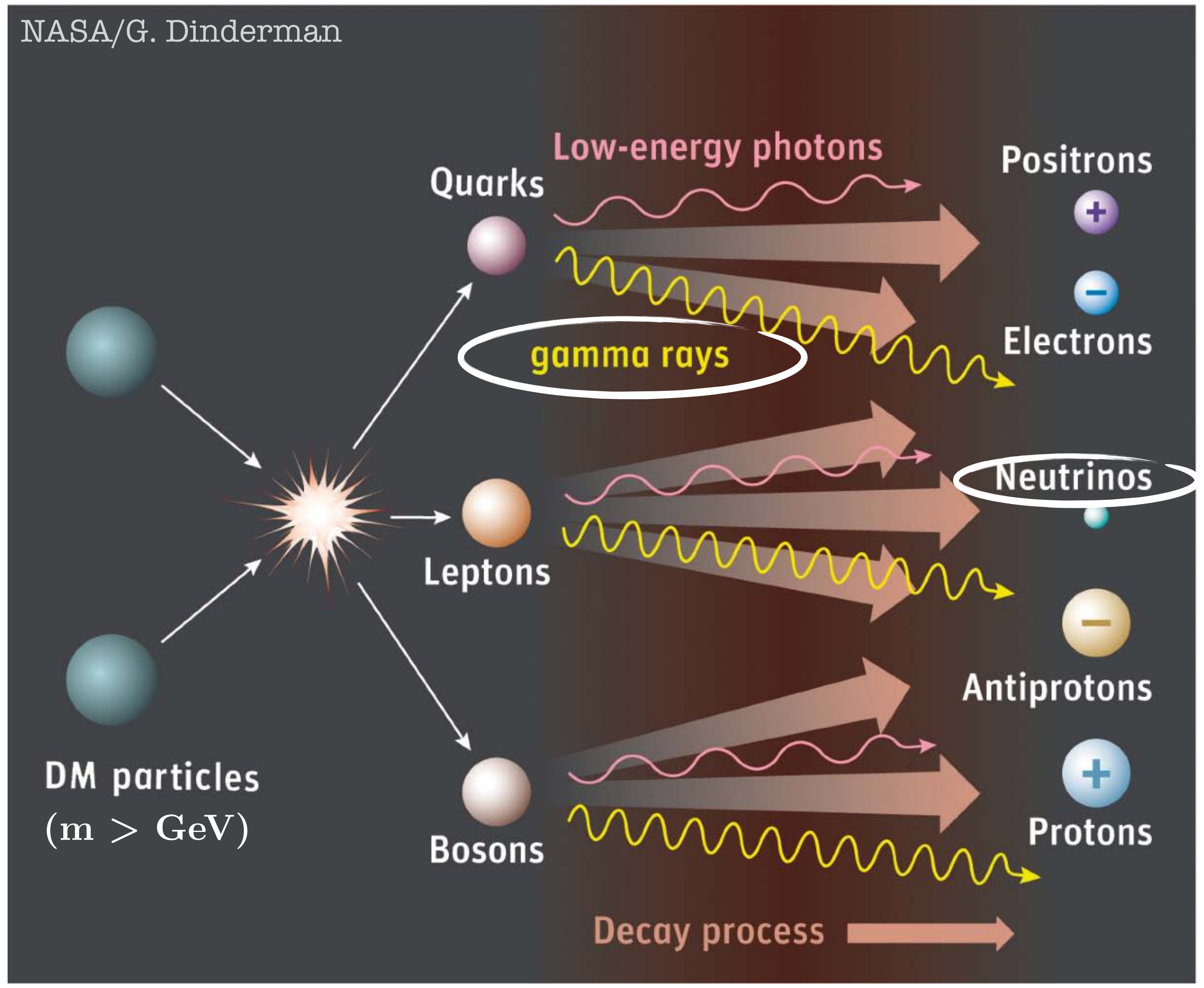
Bridge between heavy numerical simulations and experiments:

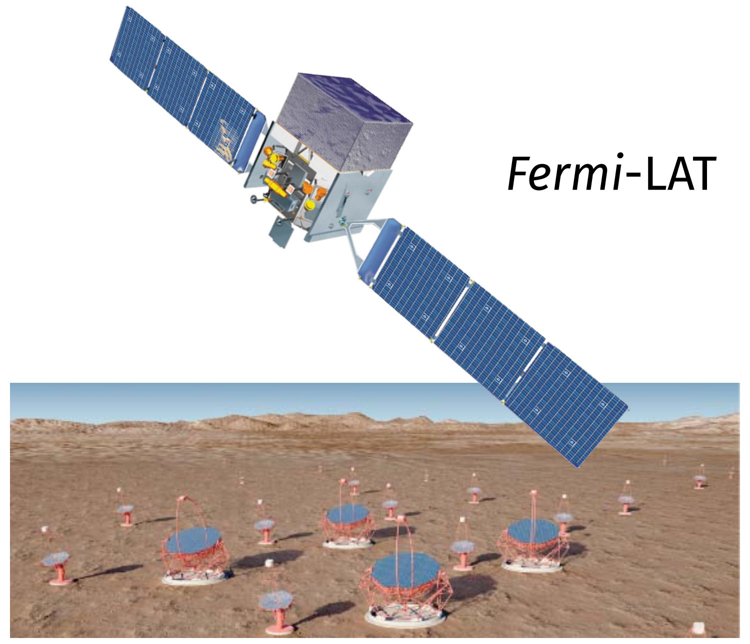
- Fast emulator to calculate J-factors/fluxes from simulation end-products down to smallest mass scales
- Explore varying simulation results in a parametric way: fast, flexible, user-friendly
- Jeans-analysis module to reconstruct dSph DM density profiles from kinematic data

Indirect DM detection in γ-rays and ν: Recap

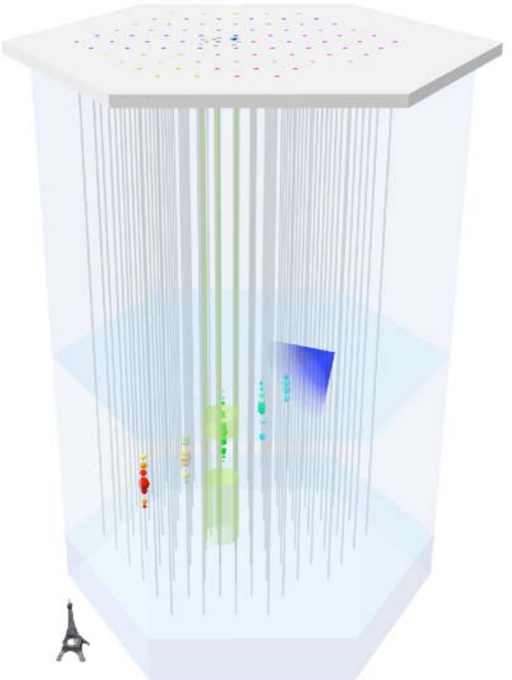
Dense & massive astrophysical DM budget → Annihilation or decay of the DM → Detectors for astrophysical γ-rays and neutrinos






Fermi-LAT

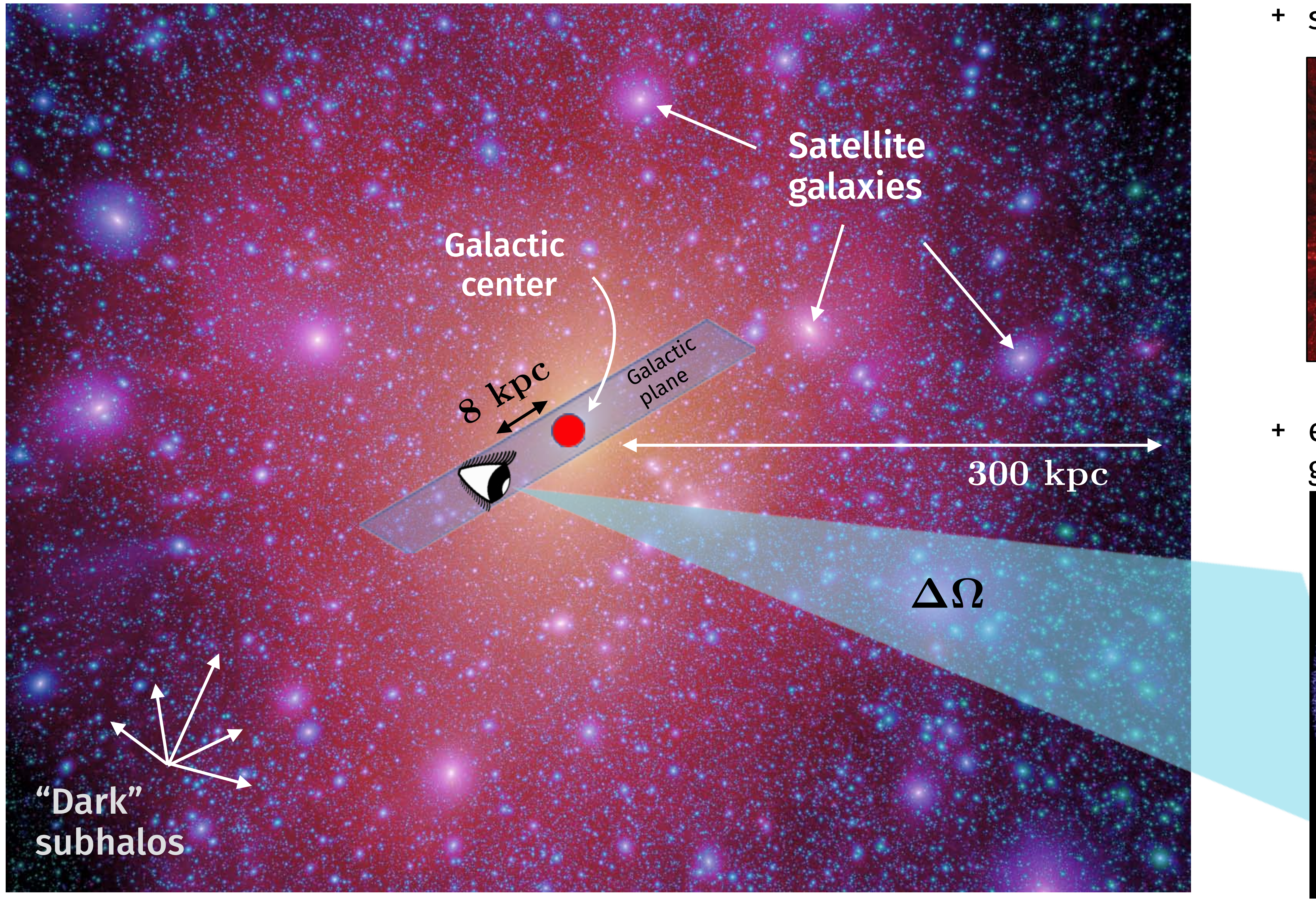
CTA


IceCube

Indirect DM detection in neutral particles: Recap

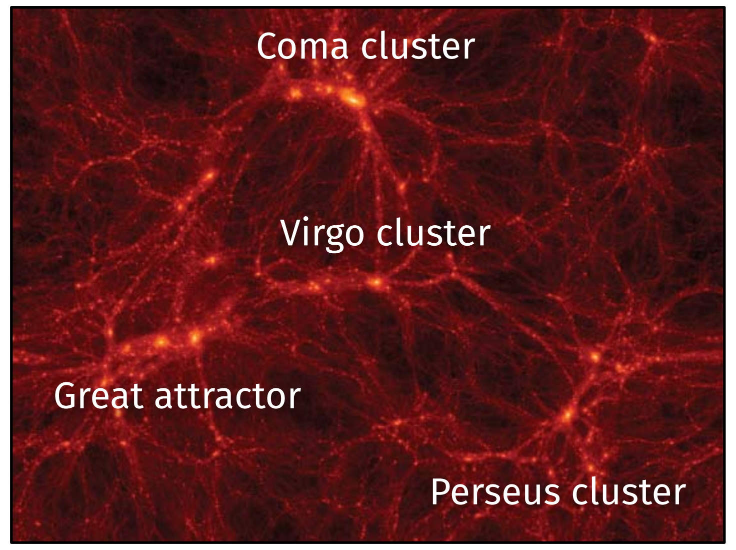
Where to look?

Massive & dense (M^2/V) vs. close ($1/d^2$) vs. little astrophysical background



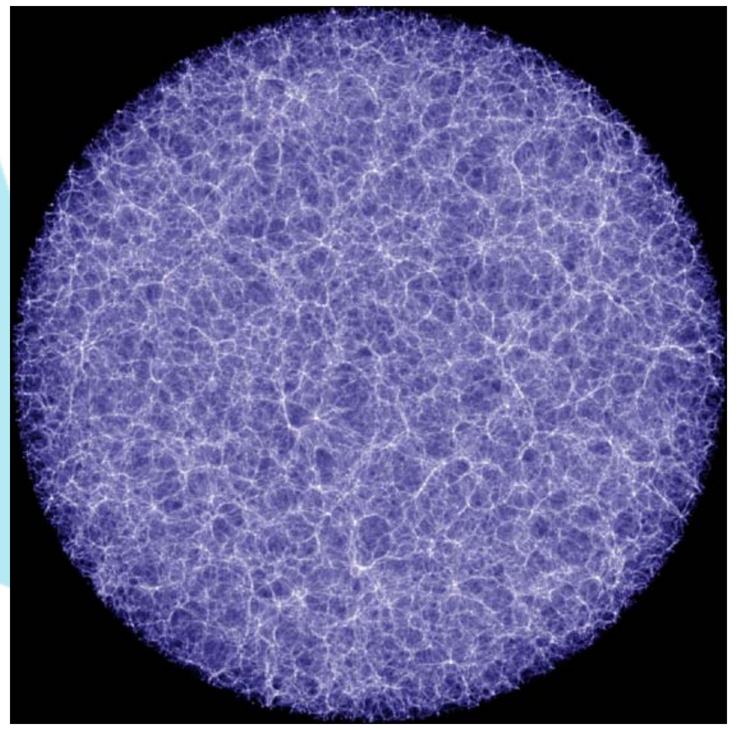
Aquarius simulation – Springel et al. (Nature, 2008)

- single galaxy clusters ($d >$ Mpc)



Gottlöber et. al. (2010)

- ensemble average of extra-galactic DM ($d >$ Gpc)



Angulo et al. (2008)

CLUMPY calculates J-factors/fluxes for all the various targets

Indirect DM detection in neutral particles: Recap

Prompt γ-ray/ν flux for single source & DM **annihilation**:

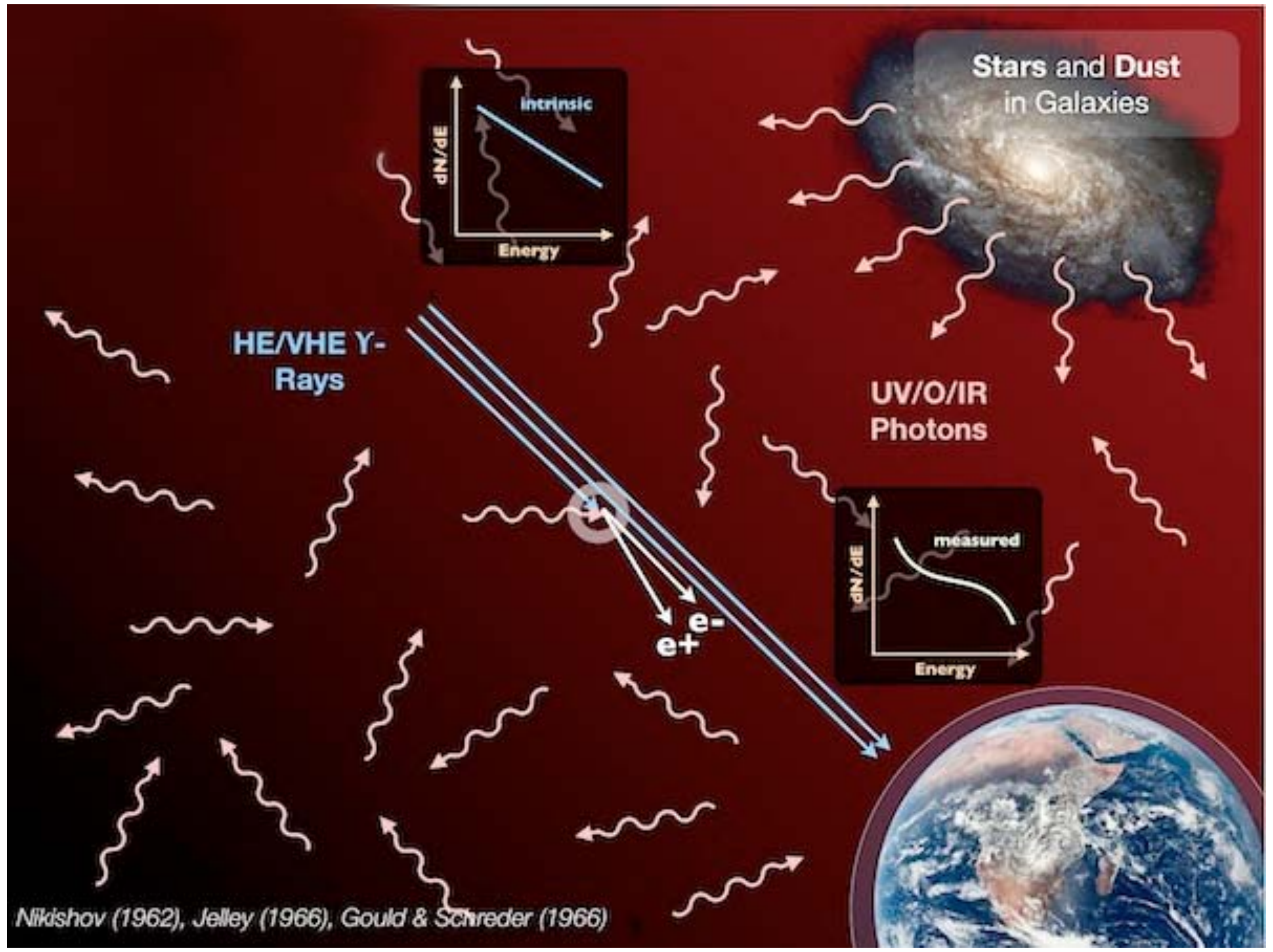
$$\frac{\mathrm{d}\Phi^{\mathrm{ann}}}{\mathrm{d}E_{\mathrm{obs}}} = \frac{1}{4\pi}\frac{\langle\sigma v\rangle}{\delta\, m_\chi^2} \times \frac{\mathrm{d}N}{\mathrm{d}E}(E) \times \int_{\Delta\Omega}\int_{l.o.s.} \rho_{\mathrm{DM}}^2 \,\mathrm{d}l\,\mathrm{d}\Omega$$

Flux = Particle physics × J : Astrophysical factor $\propto \approx \frac{1}{d^2}\frac{M^2}{V}$

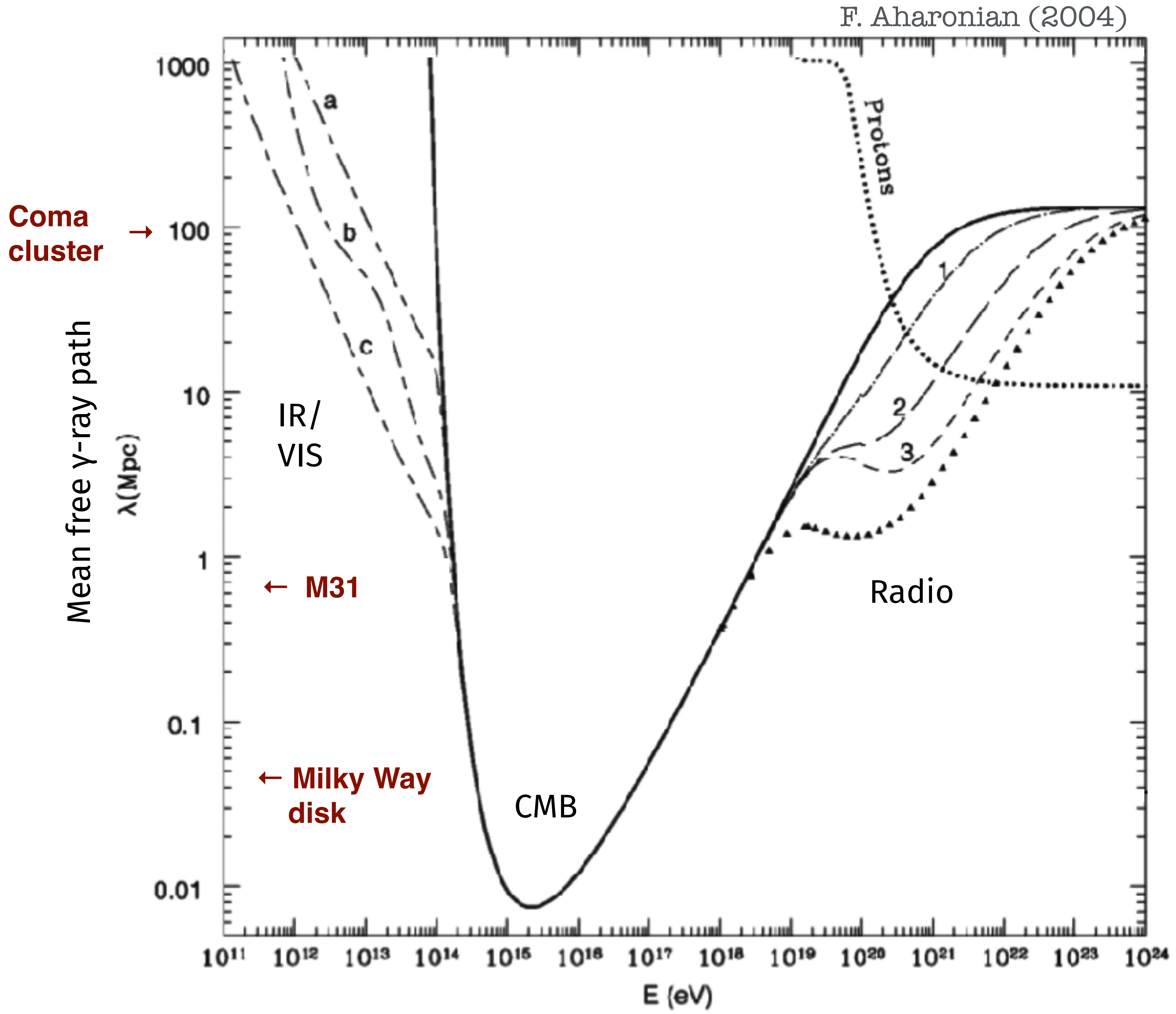
(CLUMPY can also do all calculations for DM **decay**)

J-factor main uncertainty in indirect DM searches

- γ-rays from outside the local Universe:
  - Redshifting of the γ-rays/ neutrino energy loss
  - γ-rays absorption by pair-production with photons of the extragalactic background light (EBL)



LEXI, University of Hamburg



Indirect DM detection in extragalactic γ-rays

➡ More intricate form of flux equation (single extragalactic object):

EBL absorption

$$\frac{d\Phi_\gamma^{\mathrm{ann.}}}{dE_\gamma^{\mathrm{obs}}} = \frac{1}{4\pi}\frac{\langle\sigma v\rangle}{2m_\chi^2} \times \frac{dN_\gamma}{dE_\gamma^{\mathrm{source}}}\left([1+z]\,E_\gamma^{\mathrm{obs}}\right) \times e^{-\tau(z,E_\gamma)}$$

redshift

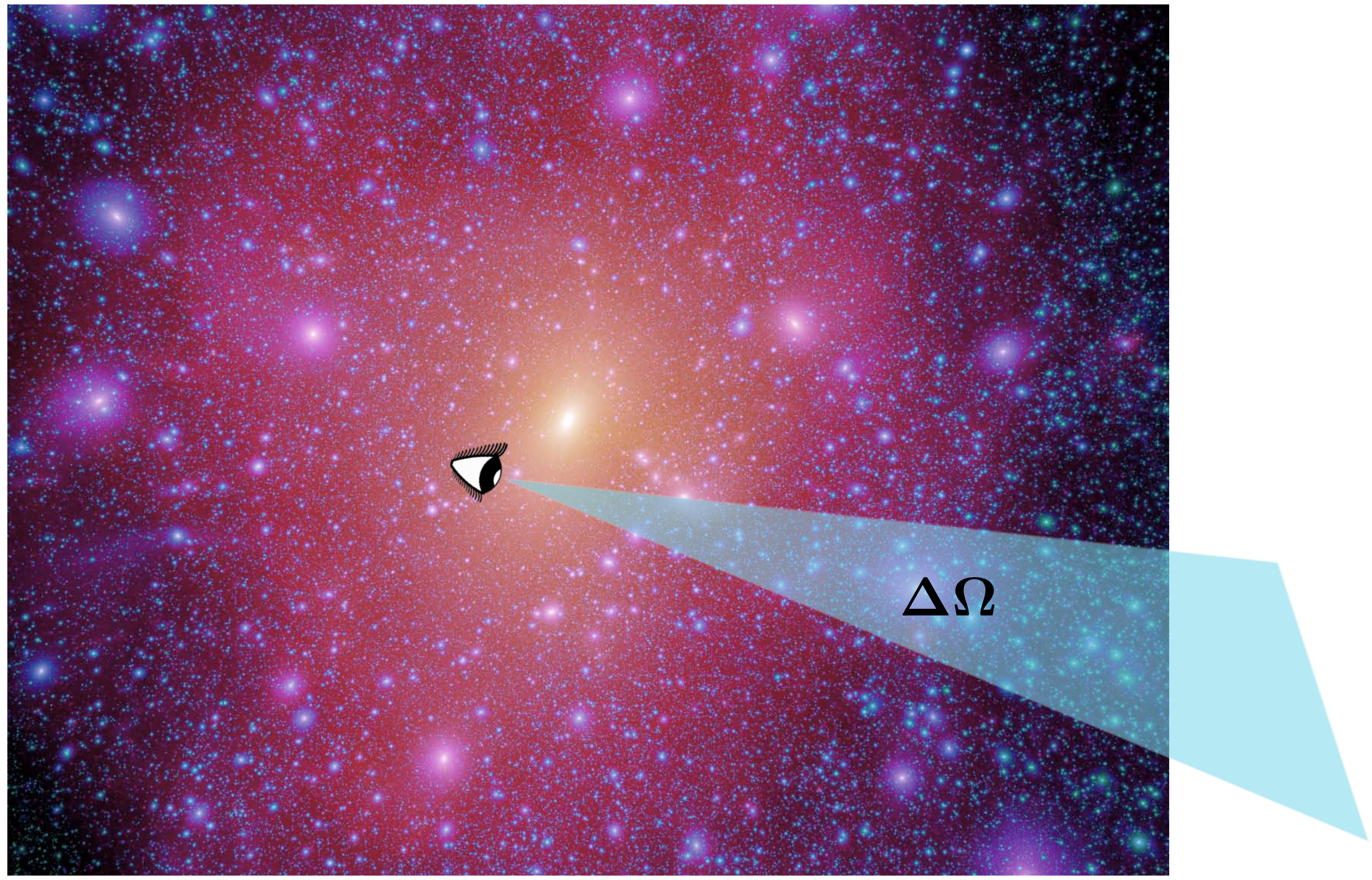
$$\times\,(1+z)^3 \int_{\Delta\Omega}\int_{l_c} \rho_{\mathrm{DM}}^2\, dl_c\, d\Omega$$

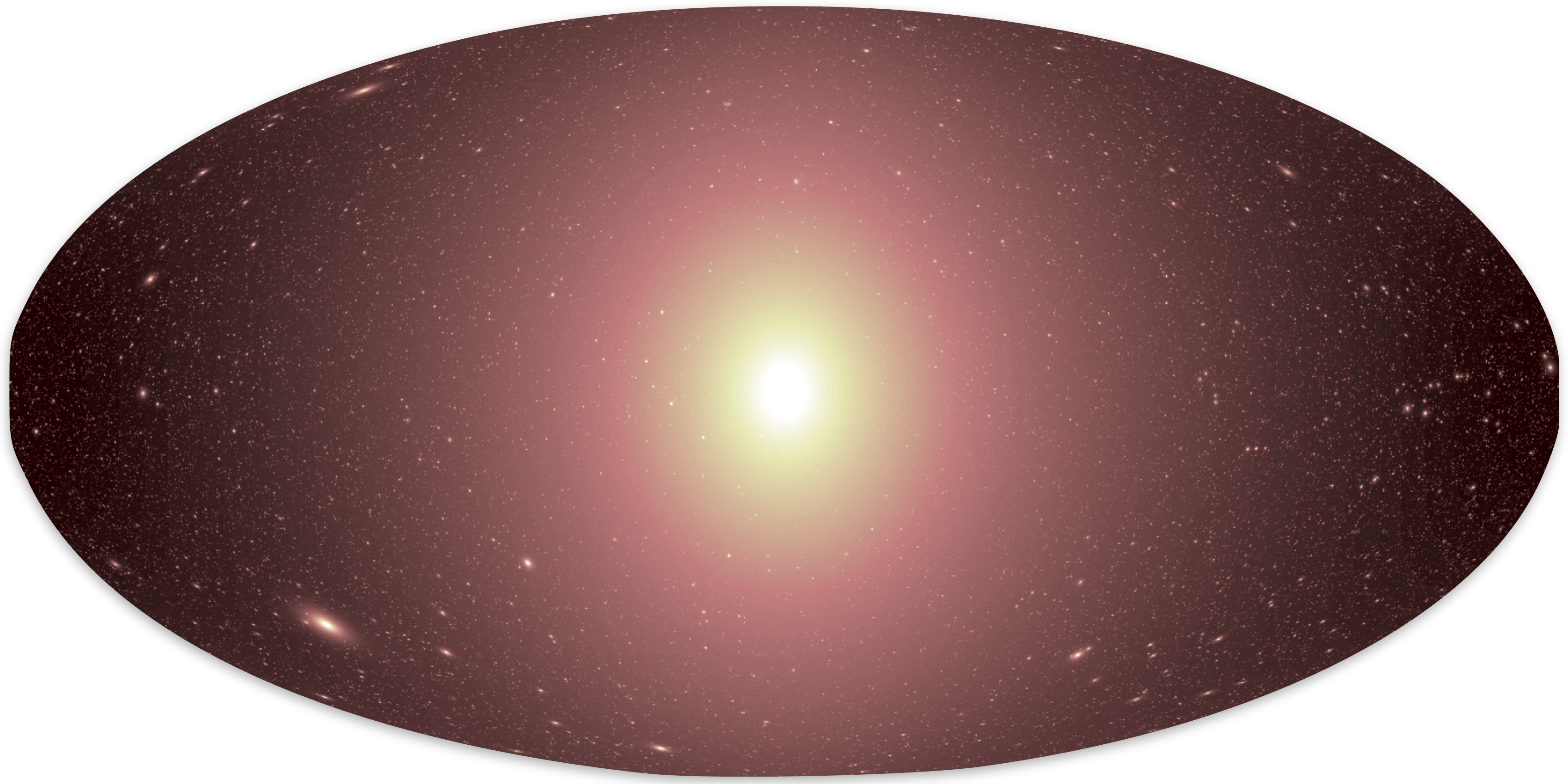
annihilation boost in smaller proper volume

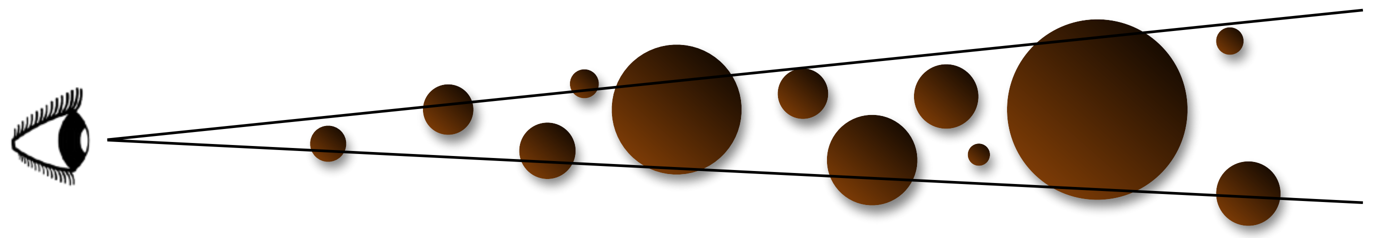
description in comoving coordinates

➡ N.B.: Separation in particle physics/astrophysics term breaks down if considering a signal originating from multiple redshift shells

CLUMPY: rationale







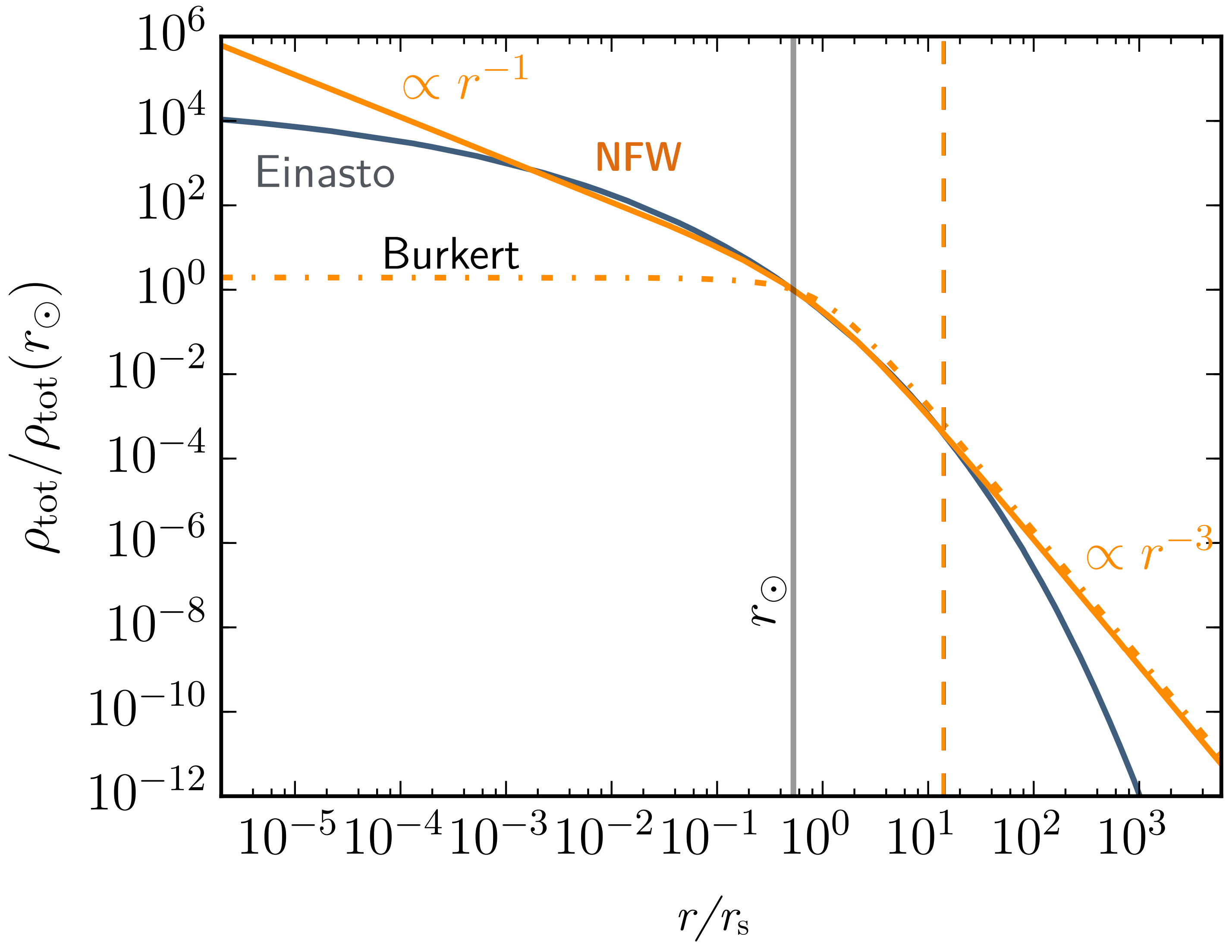
CLUMPY: *J*-factor integrator (with accounting for many subtleties)

- 1D: J-factor or fluxes integrated over solid angle $\Delta\Omega$
- 2D: Skymaps for $\mathrm{d}J/\mathrm{d}\Omega$ or $\mathrm{d}\Phi/\mathrm{d}\Omega$

(Jeans-analysis module not covered in today's tutorial)

DM halos: Recap

- In CLUMPY, use various parametric families od spherical density profiles (next release: also arbitrary numerically defined profile possible)
- Once family-parameters are fixed, **a halo is fully defined by its scale radius r_s, density at scale radius $\rho_s = \rho(r_s)$, and outer bound R_Δ.**
- (Generalisation to triaxiality (also implemented) adds six more parameters)



`kBURKERT`	Burkert (1995): $\rho\,(r\,\|\,r_0,\rho_0) = \frac{\rho_0}{\left(1+\frac{r}{r_0}\right)\times\left[1+\left(\frac{r}{r_0}\right)^2\right]}$, with $r_{-2} \approx 1.5213797068 \times r_0$ and $\rho_0 = \rho(r=0)$.
`kEINASTO`	Navarro et al. (2004), Springel et al. (2008): $\rho\,(r\,\|\,r_{-2},\rho_{-2};\alpha) = \rho_{-2}\,\exp\left\{-\frac{2}{\alpha}\left[\left(\frac{r}{r_{-2}}\right)^{\alpha}-1\right]\right\}$
`kEINASTO_N`	Merritt et al. (2006), Graham et al. (2006): $\rho\,(r\,\|\,r_e,\rho_e;n) = \rho_e\,\exp\left\{-d_n\times\left[\left(\frac{r}{r_e}\right)^{1/n}-1\right]\right\}$, with $d_n \approx 3n - \frac{1}{3} + \frac{0.0079}{n}$ (see Merritt et al. (2006)) and $r_{-2} = r_e \times \left(\frac{2n}{d_n}\right)^n$
`kZHAO`	Hernquist (1990) and Zhao (1996): $\rho\,(r\,\|\,r_s,\rho_s;\alpha,\beta,\gamma) = \frac{2^{\frac{\beta-\gamma}{\alpha}}\times\rho_s}{\left(\frac{r}{r_s}\right)^{\gamma}\times\left[1+\left(\frac{r}{r_s}\right)^{\alpha}\right]^{\frac{\beta-\gamma}{\alpha}}}$, with $r_{-2} = r_s \times \left(\frac{\beta-2}{2-\gamma}\right)^{-1/\alpha}$. Note that we use the description where $\rho_s = \rho(r_s)$.

DM halos: Recap

- The three parameters to define a halo, r_s, $\rho_s = \rho(r_s)$, and R_Δ can be replaced by:
  - Mass of the halo, M_Δ ,
  - A functional relation $R_\Delta(M_\Delta)$,
  - The halo concentration $c = r_{-2}/R_\Delta$ (r_{-2} where the log-slope of the profile is -2, fixed relation to r_s for a profile)

- Implicitly presuming a $R_\Delta(M_\Delta)$ relation makes a profile only depend on M_Δ and c.

- Additionally presuming a strict $c(M_\Delta)$ relation makes a halo uniquely be defined by its mass ("all halos of the same mass look the same")

Halo definition in terms of mass and concentration: Recap

R_Δ as function of halo mass:

$$R_\Delta(M_\Delta, z) = \left(\frac{3\, M_\Delta}{4\pi \times \Delta(z) \times \varrho_c(z)} \right)^{1/3} \times (1+z)$$

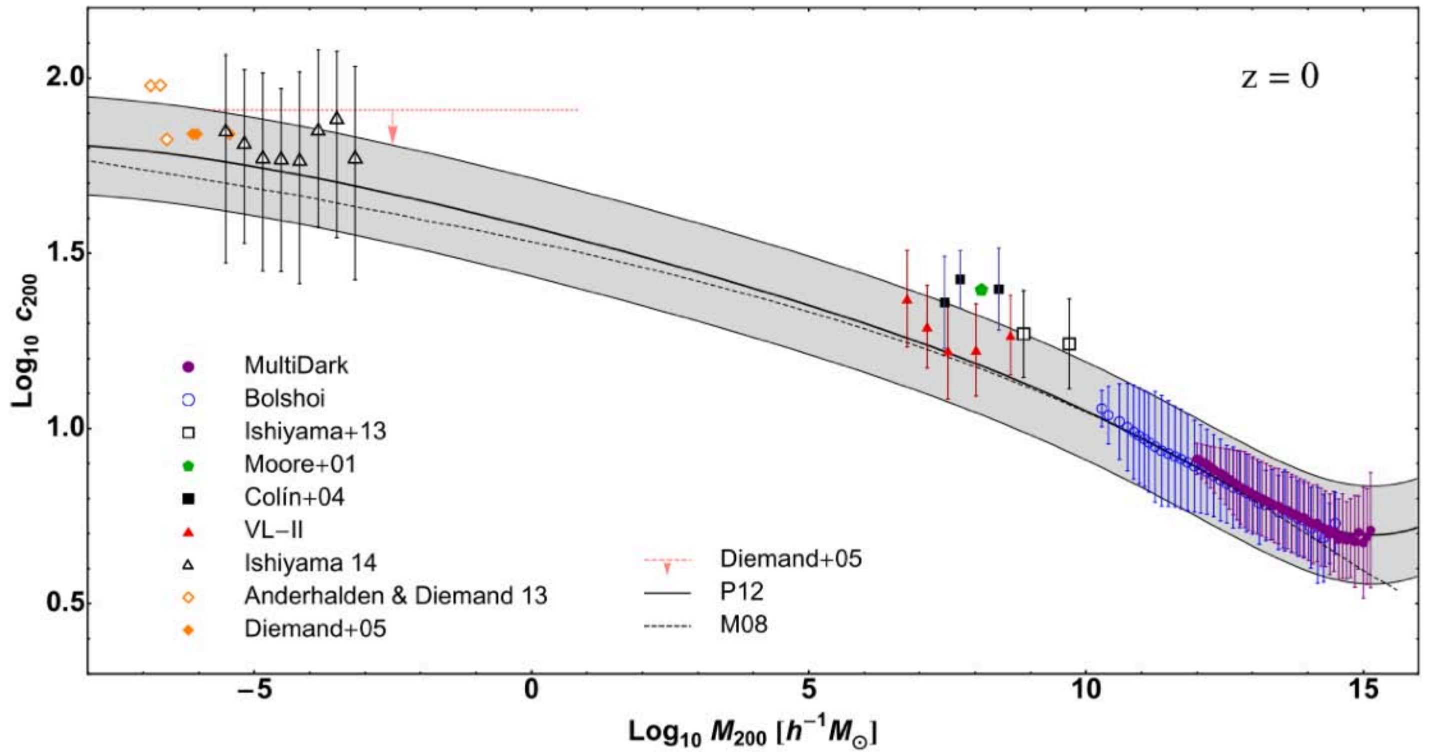
$$\Delta(z) = \text{const.},$$
$$\Delta(z) = \text{const.} \times \Omega_m(z) =: \Delta_m \times \Omega_m(z),$$
$$\Delta(z) = 18\pi^2 + 82\,[\Omega_m(z) - 1)] - 39\,[\Omega_m(z) - 1]^2.$$

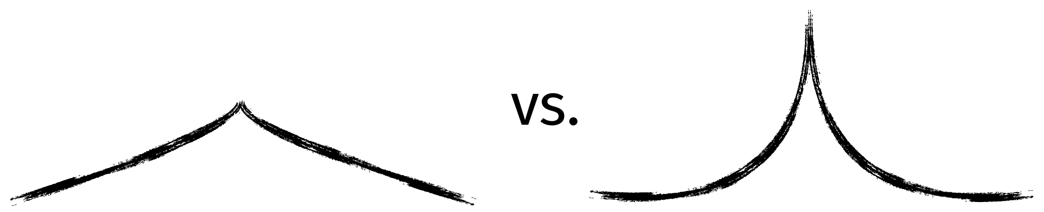
Descriptions in CLUMPY

Keyword	Description
kRHO_CRIT	$\Delta(z) = \Delta_{\rm crit}$
kRHO_MEAN	$\Delta(z) = \Delta_{\rm crit} \times \Omega_m(z)$
kBRYANNORMAN98	Using the formula by Bryan and Norman (1998) for a flat Universe

$c = r_{-2}/R_\Delta$ as function of halo mass:

Sánchez-Conde et al., 1312.1729





Concentration: measure of the "spikyness" of a halo: Defined as

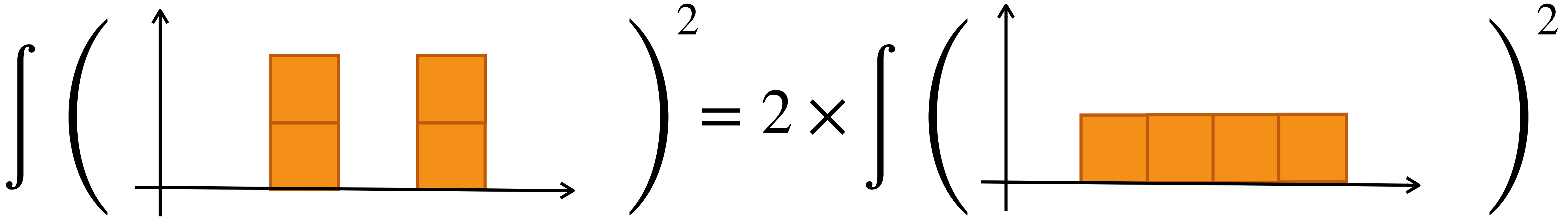
Descriptions in CLUMPY

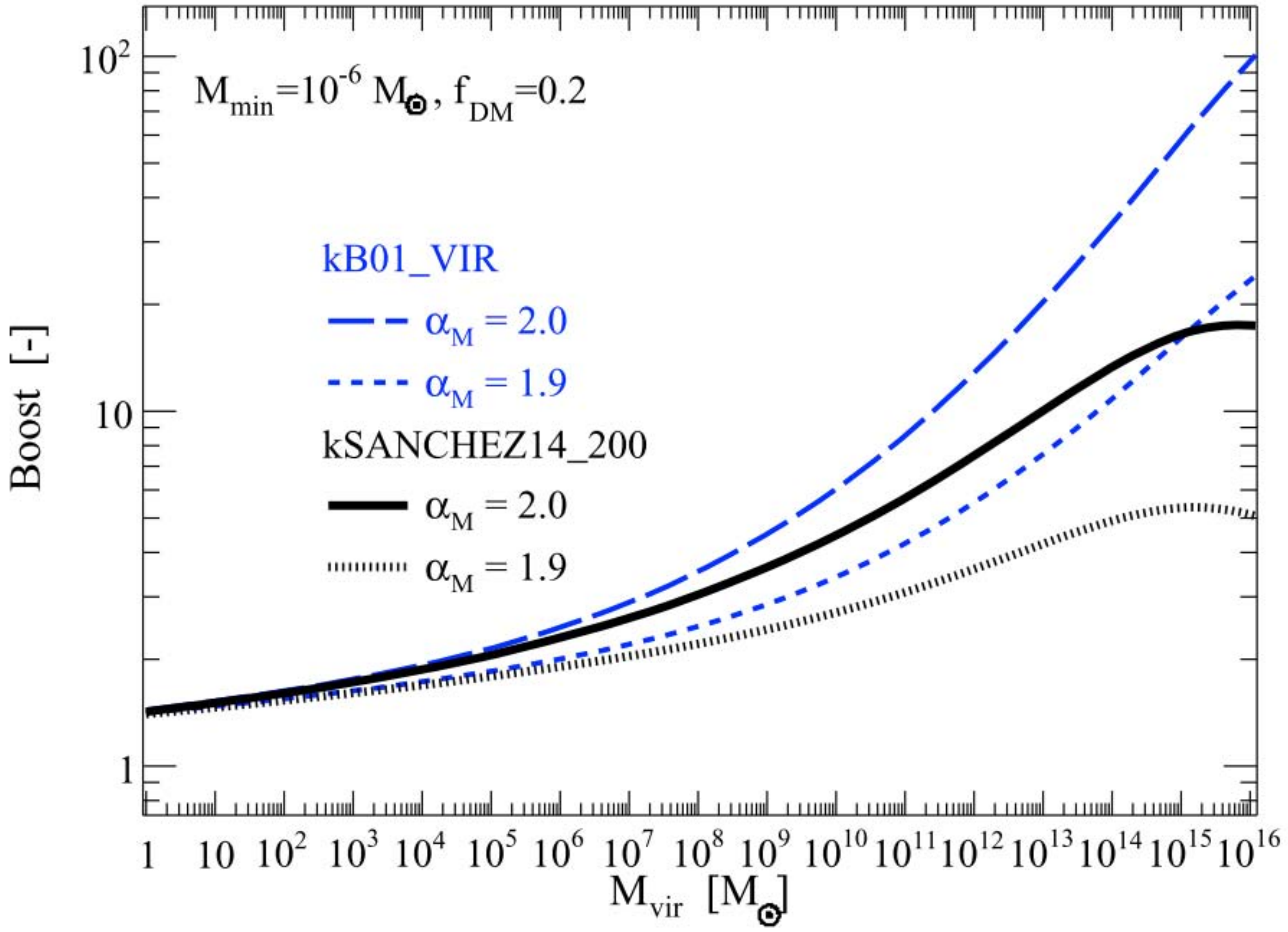
Keyword	Description
kB01_VIR	Bullock et al. (2001): extrapolation down to any mass range. $C_i = \{-4.34,\ 0.0384,\ -3.91 \times 10^{-4},\ -2.2 \times 10^{-6},\ -5.5 \times 10^{-7}\}$
kB01_VIR_RAD	Kuhlen et al (2008): adaption of Bullock et al. (2001) with radial dependence.
kCORREA15_PLANCK_200	Correa et al. (2015): Only for extragalactic haloes and Planck cosmology.
kENS01_VIR	Eke et al. (2001): extrapolation down any mass range. $C_i = \{3.14,\ -0.018,\ -4.1 \times 10^{-4}\}$
kNETO07_200	Neto et al. (2007): valid for $M > 10^8 M_\odot$. $c_{200} = 4.67 \times \left(\frac{M_{200}}{10^{14} M_\odot}\right)^{-0.11} \times (1+z)^{-1}$ [their Eq. 5].
kDUFFY08F_VIR	Duffy et al. (2008), $\Delta = \Delta_{\rm vir}(z)$: valid for $M > 10^8 M_\odot$. $c_{\rm vir} = A\left(\frac{M_{\rm vir}}{2{\times}10^{12} M_\odot}\right)^B (1+z)^C$ [see their Table 1 for A, B, C].
kDUFFY08F_200	Duffy et al. (2008), $\Delta = 200$: valid for $M > 10^8 M_\odot$.
kDUFFY08F_MEAN	Duffy et al. (2008), $\Delta = 200 \times \Omega_{\rm mean}(z)$: valid for $M > 10^8 M_\odot$.
kETTORI10_200	Ettori et al. (2010): valid for $M > 10^8 M_\odot$ $c_{200} = 10^{0.62} \times \left(\frac{M_{200}}{10^{15} M_\odot}\right)^{-0.1} \times (1+z)^{-1}$ [see their Eq. 5].
kGIOCOLI12_VIR	Giocoli et al. (2012): extrapolation down to any mass range.
kLUDLOW16_200	Ludlow et al. (2016): Only for extragalactic haloes and Planck cosmology.
kMOLINE17_200	Moliné et al. (2017), Eq. (6): improvement on Sánchez-Conde & Prada (2014) for Galactic subhalos accounting for spatial dependence of subhalos in field halos.

2. The substructure problem

Non-linear contribution from substructures: Recap

$$J = \int_{\Delta\Omega} \int_{l.o.s.} \rho_{\rm DM}^2 \, {\rm d}l \, {\rm d}\Omega$$





- Signal gets enlarged for very dense packing of mass
- Signal gets enlarged for sub-(sub-sub…)-structuring of the mass
- While integral over fractal density (= total mass) is finite, integral over square-density may diverge

Substructure boost $= \dfrac{J_{\rm smooth} + \langle J_{\rm cross-prod} \rangle + \langle J_{\rm sub} \rangle}{J_{\rm no\ subs}}$

dSph: boost $\sim 1 - 2$

Galaxy cluster: boost $\sim 10 - 100$

Averages for the substructure signal: Recap

$$J = \int_0^{\Delta\Omega}\int_{l_{\rm min}}^{l_{\rm max}} \frac{1}{l^2}\left(\rho_{\rm sm} + \sum_i \rho_{\rm cl}^i\right)^2 l^2 dl d\Omega$$

User-defined input quantity

$$J_{\rm sm} \equiv \int_0^{\Delta\Omega}\int_{l_{\rm min}}^{l_{\rm max}} \rho_{\rm sm}^2 dl d\Omega$$

$$J_{\rm cross-prod} \equiv 2\int_0^{\Delta\Omega}\int_{l_{\rm min}}^{l_{\rm max}} \rho_{\rm sm}\sum_i \rho_{\rm cl}^i dl d\Omega$$

$$J_{\rm subs} \equiv \int_0^{\Delta\Omega}\int_{l_{\rm min}}^{l_{\rm max}} \left(\sum_i \rho_{\rm cl}^i\right)^2 dl d\Omega$$

exact realisation (mass and position) of DM substructures not needed

Do not resolve trillions of substructures, but calculate the average signal over ensemble quantities

$$\langle J_{\rm cross-prod}\rangle = 2\int_0^{d\Omega}\int_{l_{\rm min}}^{l_{\rm max}} dl d\Omega\, \rho_{\rm sm}\langle\rho_{\rm subs}\rangle$$

$$\langle\rho_{\rm subs}\rangle = f_{\rm subs}\, M_{\rm halo}\, \frac{d\mathscr{P}_V}{dV}(r)$$

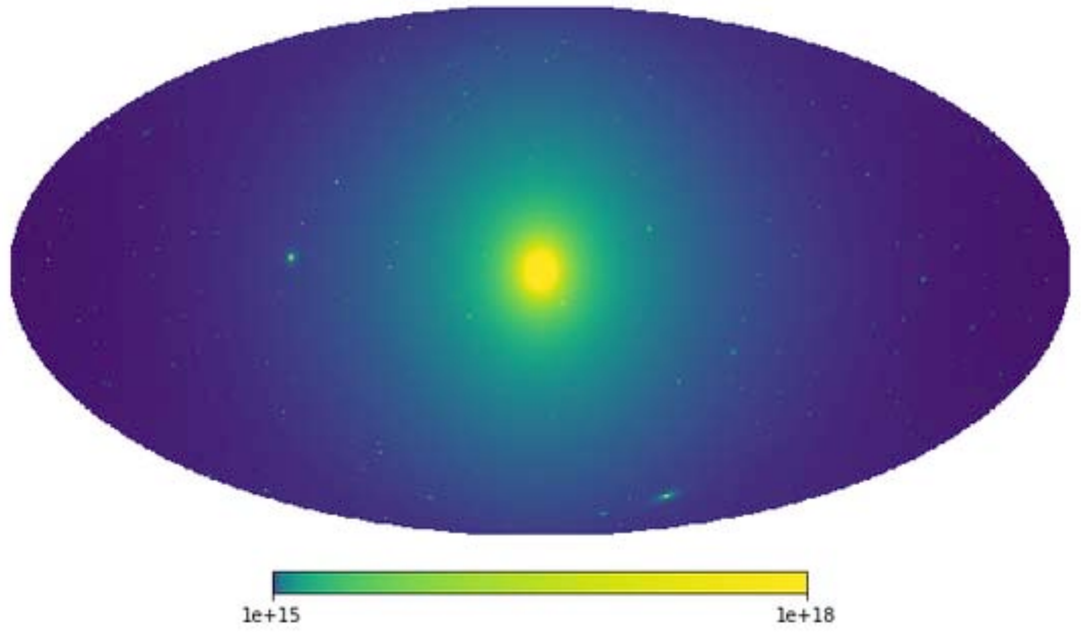
$$\langle J_{\rm subs}\rangle = \int_0^{d\Omega}\int_{l_{\rm min}}^{l_{\rm max}}\int_{M_{\rm min}}^{M_{\rm max}}\int_{c_{\rm min}}^{c_{\rm max}} dl d\Omega dM dc \frac{d^3N}{dV dM dc}\mathscr{L}(M,c)$$

$$\mathscr{L}(M,c) = \int_{V_{\rm cl}} dV\,(\rho_{\rm cl})^2$$

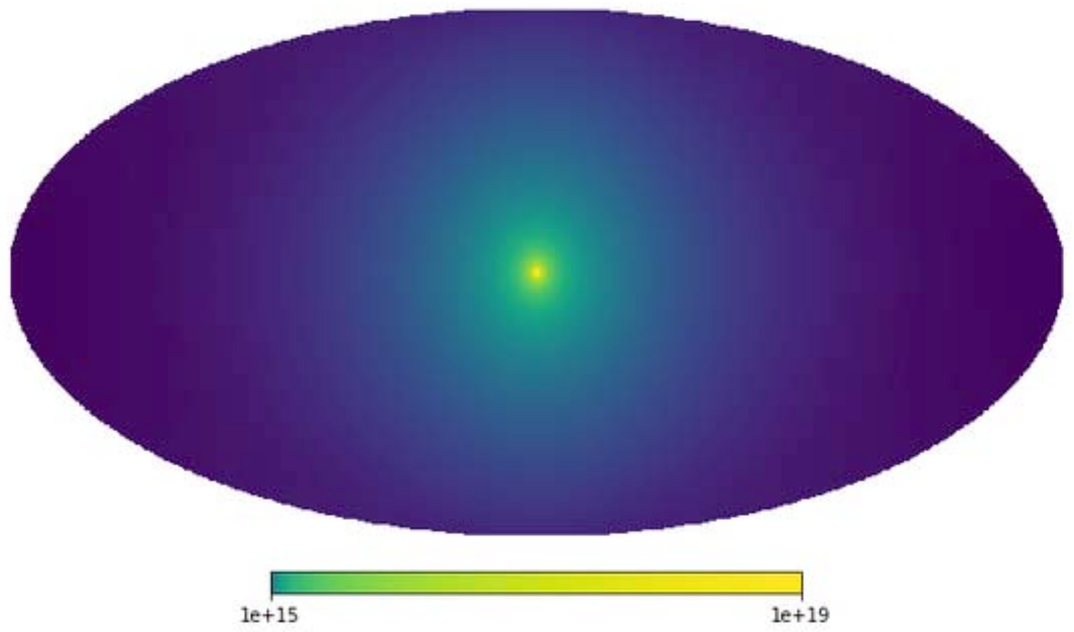
$$\rho_{\rm sm} = \rho_{\rm tot} - \langle\rho_{\rm subs}\rangle$$

Four contributions to the J-factor

Total signal from DM annihilation

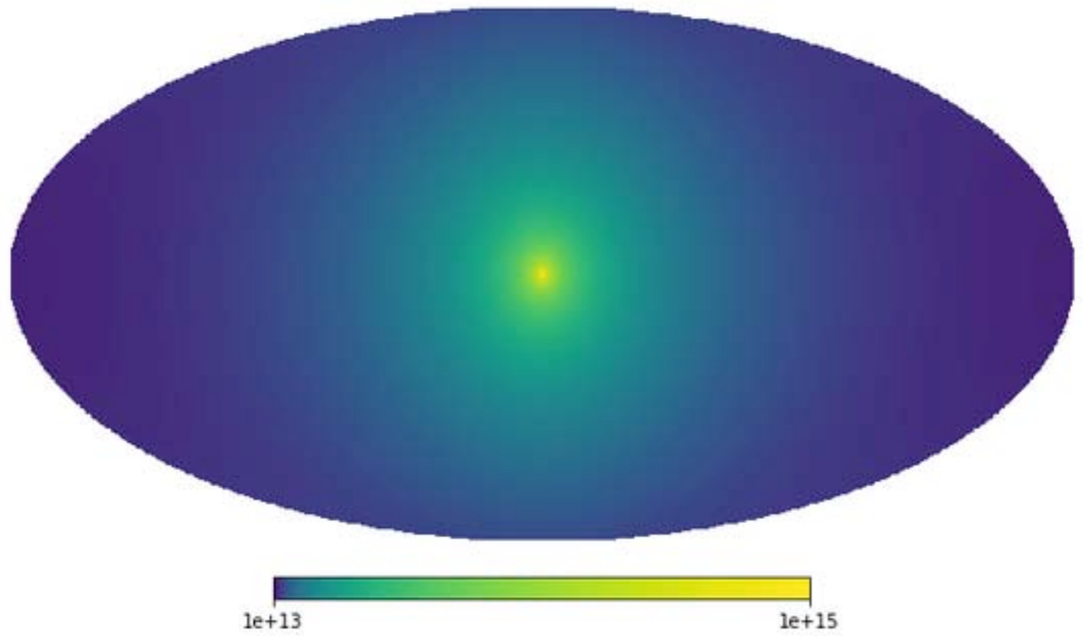


=



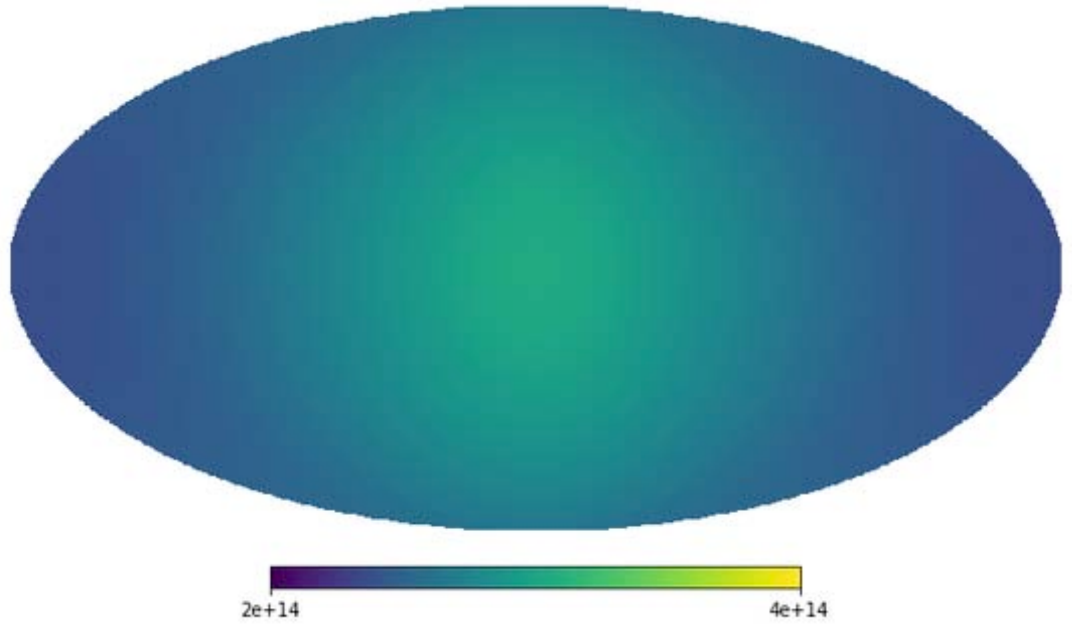
$J_{\rm sm}$

+



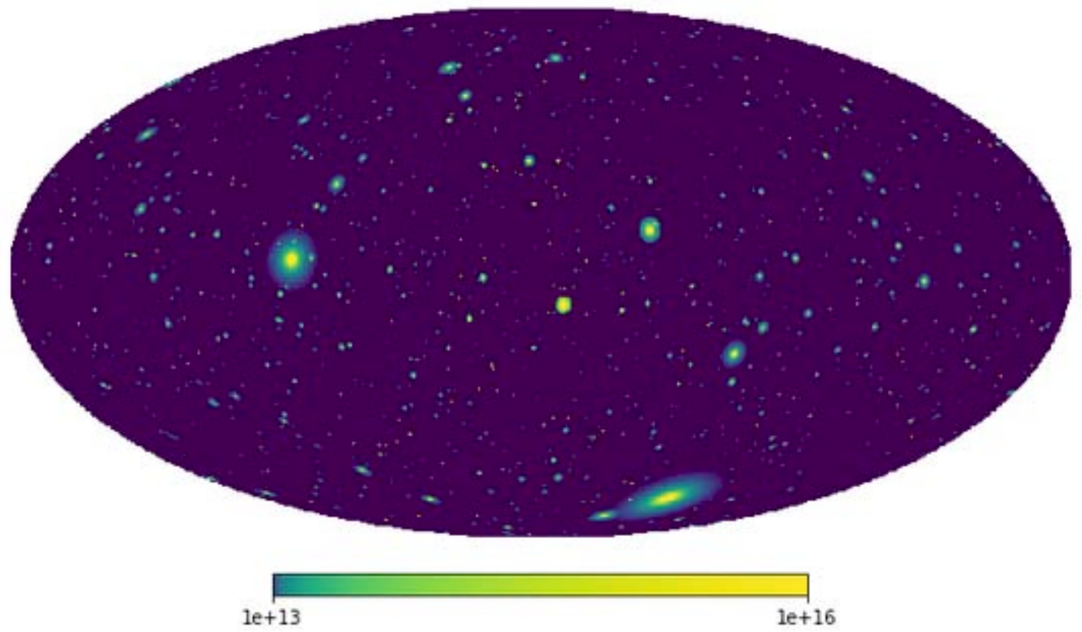
$\langle J_{\rm cross-prod} \rangle$

+



$\langle J_{\rm subs} \rangle$

+



$J_{\rm drawn}$

resolved substructures,
see later

The $<J_{\rm subs}>$ term

User-defined input quantity

$$\langle J_{\rm subs}\rangle = \int_0^{d\Omega}\int_{l_{\rm min}}^{l_{\rm max}}\int_{M_{\rm min}}^{M_{\rm max}}\int_{c_{\rm min}}^{c_{\max}} dl d\Omega dM dc \; \frac{d^3N}{dVdMdc}\,\mathscr{L}(M,c)$$

$$\frac{d^3N}{dVdMdc} = N_{\rm subs}\,\frac{d\mathscr{P}_V}{dV}(r)\times\frac{d\mathscr{P}_M}{dM}(M)\times\frac{d\mathscr{P}_c}{dc}(M,r,c)$$

- $d\mathscr{P}_V/\,dV$: Probability to fin clump at some radial position in the host halo: usually flatter than total DM density
- $d\mathscr{P}_M/\,dM \sim M^{-\alpha_M}$: probability density of substructure mass: independent of position in halo
- $N_{\rm subs} = f_{\rm subs}\,M_{\rm halo} \Big/ \int_{M_{\rm min}}^{M_{\rm max}} dM\,M\,\frac{d\mathscr{P}_M}{dM}$: total number of substructures in halo
- $d\mathscr{P}_c/\,dc$: Log-normal distribution around mean $\bar{c}(M,r)$

In worst/naive case, $5^{\rm N\ sub-levels}$ dimensional integral to calculate (slow…)

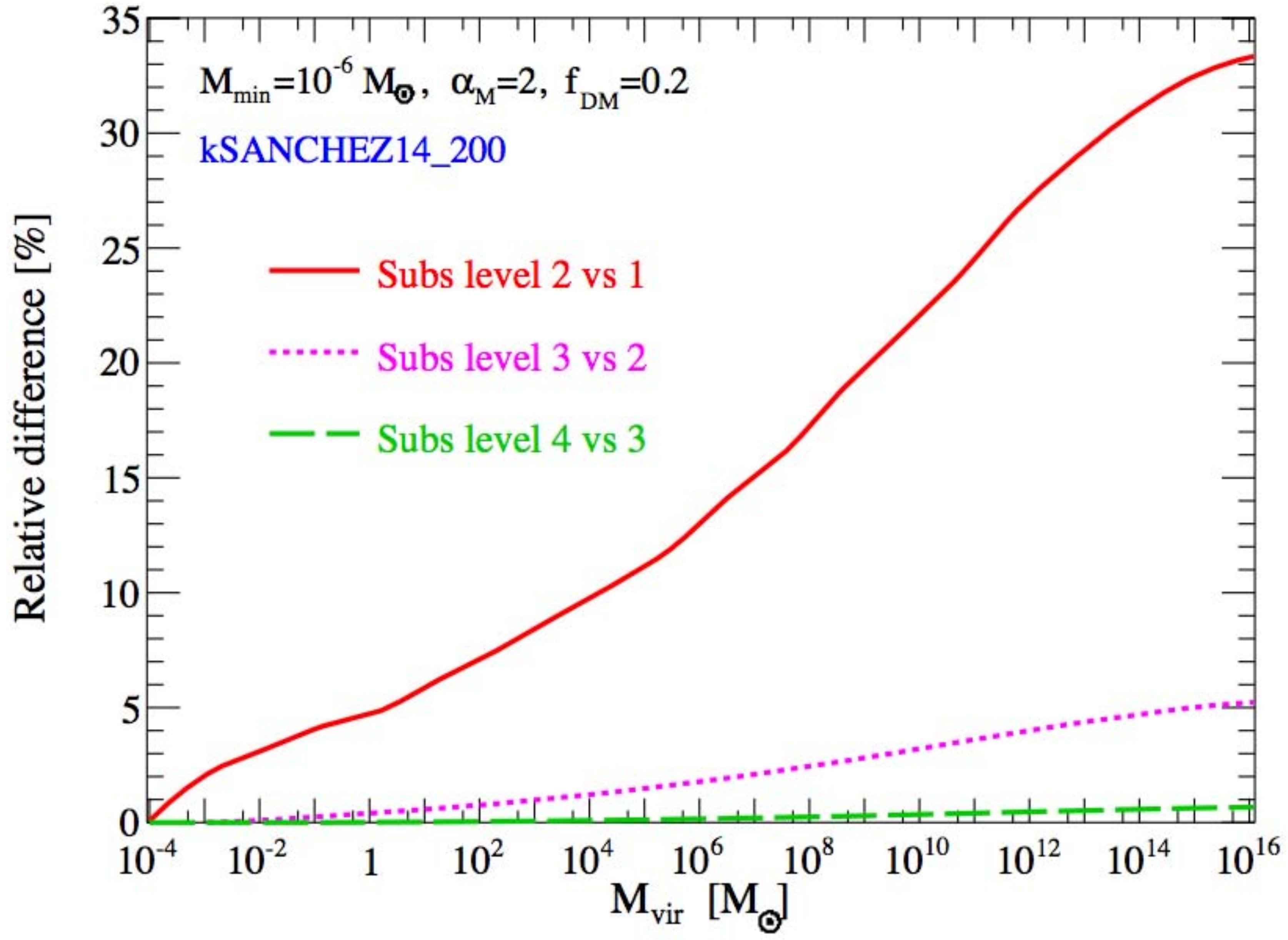
Multi-level substructures

- Additionally, the signal boost from sub-substructures can be computed in an iterative way:

$$\mathcal{L}_n(M) = \mathcal{L}_{\rm sm}(M) + \mathcal{L}_{\text{cross-prod}}(M) + N_{\rm sub}(M) \int_{M_{\rm min}}^{M_{\rm max}(M)} \mathcal{L}_{n-1}(M') \frac{d\mathcal{P}}{dM'}(M')dM'$$

$$\mathcal{L}_{\rm sm}(M) \equiv \int_{\rm V_{cl}} \left[\rho_{\rm cl}^{\rm sm}(M)\right]^2 \, dV;$$

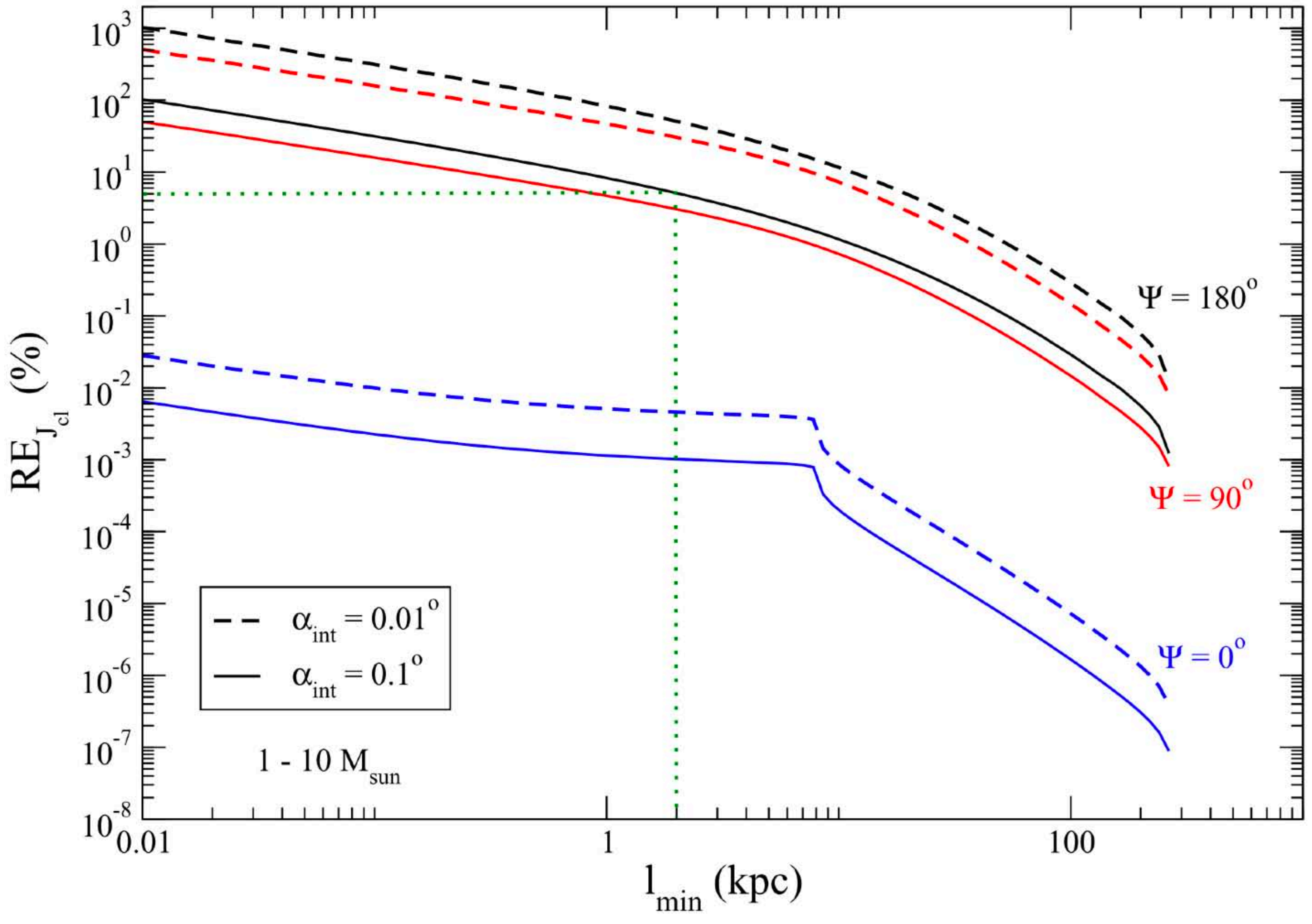
$$\mathcal{L}_{\text{cross-prod}}(M) \equiv 2 \int_{\rm V_{cl}} \rho_{\rm cl}^{\rm sm}(M) \, \langle \rho_{\rm subs}(M) \rangle \, dV$$



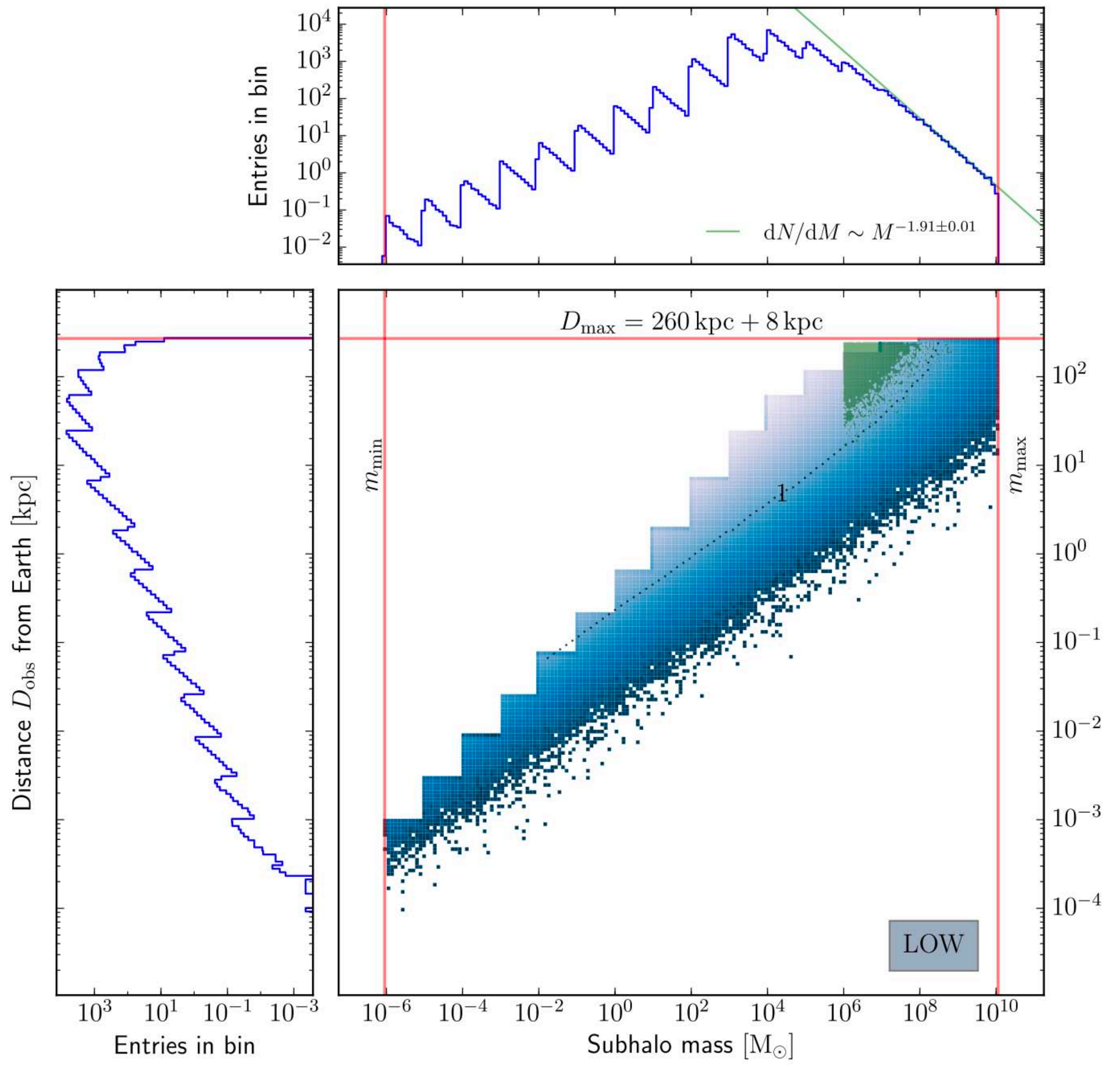
- Despite many optimizations of the integration, it is not computationally not possible to compute multi-level substructures with scatter in the concentration

2D skymaps: Resolving the substructures with the Relative-Error approach

- Resolving signal from all substructures in a halo ($N_{subs} \sim 10^{14\text{-}15}$ for m_{min}=10^{-6} M_{sol} and α_M=2) impossible
- Applying a mass-cut also not convenient: Close-by light clumps may shine brighter than far-away heavy ones
- Choose distance below which to resolve as function of mass: $l_{min}(M)$
- $l_{min}(M)$ determined by the maximum allowed relative error w.r.t. the varying flux if all substructures were resolved:



$$RE_{J_{clumps}} = \frac{\sqrt{N_{cl}}\ \sigma_{1\,cl}}{N_{cl}\langle J_{1\,cl}\rangle + J_{smooth}}$$



5. Commanding the code

```
   if (is_simple_interp) printf("        ... Fill interpolation function (%d lin-steps) ...\n", n_base);
   else printf("        ... Fill interpolation function (%d log-steps) ...\n", n_base);

   j_1D_base.assign(n_base, 1.e-40);
   phi_base.assign(n_base, 1.e-40);
   iphi_inbase.assign(n_1D, -1);
   double delta_phi_base;
   if (is_simple_interp) delta_phi_base = (phi_max - phi_min) / double(n_base - 1);
   else delta_phi_base = pow(phi_max / alpha_quad_start, 1. / double(n_base - 1));


   for (int i = 0; i < n_base; ++i) {
      if (is_simple_interp) phi_base[i]  = phi_min + i * delta_phi_base;
      else phi_base[i] = alpha_quad_start * pow(delta_phi_base, i);
      double jopt = 1.e-40;
      if (switch_j == 0) {
         if (f_dm > 1.e-3)
            jopt = jsmooth_mix(mtot, par_tot, phi_base[i], theta_1D, lmin, lmax, eps, f_dm, par_dpdv);
         else
            jopt = jsmooth(par_tot, phi_base[i], theta_1D, lmin, lmax, eps);
      } else if (switch_j == 1) {
         // N.B.: we have to take into account all mass decades
         for (int k = 0; k < n_mass; ++k) {
            if (l_crit[k] < lmax)
               jopt += jsub_continuum(ntot_subs, par_dpdv, phi_base[i], theta_1D,
                                      l_crit[k], lmax,  par_subs, m1[k], m2[k]);
         }
      } else if (switch_j == 2) {
         // N.B.: we have to take into account all mass decades
         for (int k = 0; k < n_mass; ++k) {
            if (l_crit[k] < lmax)
               jopt += frac_nsubs_in_m1m2(&par_subs[8], m1[k], m2[k], gSIM_EPS)
                       * jcrossprod_continuum(mtot, par_tot, phi_base[i], theta_1D,
                                              l_crit[k], lmax, eps, f_dm, par_dpdv);
         }
      }
      if (jopt == 0.) jopt = 1.e-40;
      j_1D_base[i] = jopt;
   }
   // Set indices for phi_base[iphi_inbase] for phi_tab[i]
   // Search (only once) for interpolation indices for angles
   for (int i = 0; i < n_1D; ++i)
      iphi_inbase[i] = TMath::BinarySearch(n_base, &phi_base[0], phi_tab[i]);
}

if (is_interpolate) printf("        ... and interpolate for %d l.o.s. directions ...\n", n_1D);
```

CLUMPY modules

- Clumpy has 4 modules + several helper routines:
  - **-g** ➔ run on the Galactic halo only (plus optionally list of halos)
  - **-h** ➔ run on (a list of) halos (not the Gal. halo)
  - **-e** ➔ run on extragalactic DM
  - **-s** ➔ statistical Jeans-analysis

  } **The main modules**

  - **-o** ➔ export the output FITS file into a different format
  - **-z** ➔ gamma and neutrino spectra
  - **-f** ➔ append extension with fluxmaps to existing FITS file

  } **Auxiliary/additional routines**

- List is directly displayed when typing (if CLUMPY installed and $CLUMPY environment variable set):

```
> $CLUMPY/bin/clumpy
```

CLUMPY designed for self-explanatory guidance (no manual should be needed: Let us know if achieved!)

CLUMPY modules

- Every module consists of several subroutines to call. For example, the galactic module:
  - **-g0** ➔ M(r)
  - **-g1** ➔ rho$_{sm}$+<sub>(r)
  - **-g2** ➔ J$_{sm}$+<sub>(alpha$_{int}$)
  - **-g3** ➔ J$_{sm}$+<sub>(theta)
  - **-g4** ➔ J$_{sm}$+<sub>+list(theta)
  - **-g5** ➔ 2D-skymap J$_{sm}$+<sub>
  - **-g6** ➔ 2D-skymap J$_{sm}$+<sub>+list
  - **-g7** ➔ 2D-skymap J$_{sm}$+sub
  - **-g8** ➔ 2D-skymap J$_{sm}$+sub+list

- The submodules are displayed when executing CLUMPY followed by the module flag:

```
> $CLUMPY/bin/clumpy -g
```

How to execute CLUMPY

- Clumpy is by default executed by adding after calling the binary (`$CLUMPY/bin/clumpy`):
  - The submodule to call, prepended by a single dash: `-g1, -h2, -e5`,...
  - A parameter file prepended by `-i` or `--infile`: `-i param_file.txt`

- There are some additional command line options:
  - **A default parameter file for a submodule is created with `-D` (or short, e.g., `-g1D`).**
    Force creating file with all CLUMPY parameters (also unused for specific module with an additional `--all`)
  - Individual parameters (normally listed in parameter file) can also be called via the command line via.
    `--parameter_name=value.` Command line input overwrites parameter file content.
  - Use `-p` to prevent ROOT display (batch mode) or `-d` to prevent file writing (display mode)
  - Order of input does not matter, except submodule necessary to be first option.

First CLUMPY run (quick start tutorial)

- Then execute your first run by first creating a default parameter file:

```
> $CLUMPY/bin/clumpy -g1 -D
```

This will create the file `clumpy_params_g1.txt`

- Then use this just created parameter file to really execute the submodule:

```
> $CLUMPY/bin/clumpy -g1 -i clumpy_params_g1.txt
```

- This should create an output file and with ROOT linked, also a pop-up graphic.

- Switch on a linear x-axis grid by overwriting the corresponding parameter value in the file on the command line:

```
> $CLUMPY/bin/clumpy -g1 -i clumpy_params_g1.txt  --gSIM_IS_XLOG=0
```

Of course, you can alternatively also change the `gSIM_IS_XLOG` value in the parameter file.

CLUMPY parameters

- In the parameter file, parameters are sorted after topics and appended by short explanation:

```
# CLUMPY, version v3.0.1_ID5ac7986
# Standard parameter file for simulation mode g7
# Execute with $CLUMPY/bin/clumpy -g7 -i clumpy_params_g7.txt

# Variable name                 Unit          Value            (Format)          (Comment)

#------------------------------------------------------------------------------
# Cosmological parameters
#------------------------------------------------------------------------------
gCOSMO_DELTA0                   [-]           200              <float>           Overdensity factor at z=0;-1 for kBRYANNORMAN98
gCOSMO_FLAG_DELTA_REF           [-]           kRHO_CRIT        <string>          Reference density of the overdensity factor

#------------------------------------------------------------------------------
# Dark Matter global parameters
#------------------------------------------------------------------------------
gDM_LOGCDELTA_STDDEV            [-]           0                <float>           log-std. deviation of subhalo scattering
gDM_RHOSAT                      [Msol/kpc^3]  1e+19            <float>           Dark Matter saturation density
gDM_SUBS_MMIN                   [Msol]        1e-06            <float>           Smallest subclump mass
gDM_SUBS_MMAXFRAC               [-]           0.01             <float>           Biggest subclump mass, fraction of host mass
gDM_SUBS_NUMBEROFLEVELS         [-]           1                <integer>         Number of multilevel substructures

#------------------------------------------------------------------------------
# Milky-Way DM total and clump parameters (see profiles.h and clumps.h)
#------------------------------------------------------------------------------
gMW_TOT_FLAG_PROFILE            [-]           kEINASTO         <string>          Profile of total Galactic DM halo
gMW_TOT_SHAPE_PARAMS_0          [-]           0.17             <float>           Shape parameter 1 of total Gal. DM halo profile
```

CLUMPY parameters

- The full parameter list is also on the website at http://clumpy.gitlab.io/CLUMPY/parameters.html:

9. Input parameters: units and definition

9.1. Cosmological parameters

Parameter	Unit	Comment
gCOSMO_DELTA0	–	Overdensity factor at $z=0$; -1 for original kBRYANNORMAN98 description
gCOSMO_FLAG_DELTA_REF	–	Reference density of the overdensity factor, see Overdensity \Delta(z) descriptions for possible keywords
gCOSMO_HUBBLE	–	Present-day ($z=0$) normalized Hubble expansion rate $h = H_0/(100\mathrm{km\,s^{-1}\,Mpc^{-1}})$
gCOSMO_N_S	–	Scalar spectral index n_s of primordial perturbations
gCOSMO_OMEGA0_M	–	Present-day Dark and Baryonic Mass, $\Omega_{\mathrm{m},0} = \varrho_{\mathrm{m},0}/\varrho_{\mathrm{c},0}$
gCOSMO_OMEGA0_B	–	Present-day Baryonic Mass, $\Omega_{\mathrm{b},0} = \varrho_{\mathrm{b},0}/\varrho_{\mathrm{c},0}$
gCOSMO_OMEGA0_K	–	Curvature $\Omega_{\mathrm{k},0}$, only to be used with -e0 option.
gCOSMO_SIGMA8	–	Fluctuation amplitude σ_8 at $8\,h^{-1}\,\mathrm{Mpc}$
gCOSMO_T0	K	Present-day CMB photon temperature, T_0
gCOSMO_TAU_REIO	–	Reioization optical depth, τ_{reio}
gCOSMO_WDE	–	Dark energy equation of state exponent

9.2. Dark Matter global parameters

Parameter	Unit	Comment
gDM_IS_IDM	–	If True, switches on self-interacting DM halo mass function cut-off from Moliné et al. (2016).

N.B.: More complex input, e.g., the definition of halo properties or nodes of a numeric profile are outsourced to a separate file (called by the parameter file)

CLUMPY keywords

- Some parameters do not take numbers or free strings (file names), but specific keywords out of a list
- Keyword parameter values are indicated starting with a small k, e.g., `kGAMMA`.
- The available keyword values for the respective parameters are also listed on the website: http://clumpy.gitlab.io/CLUMPY/enumerators.html, e.g.:

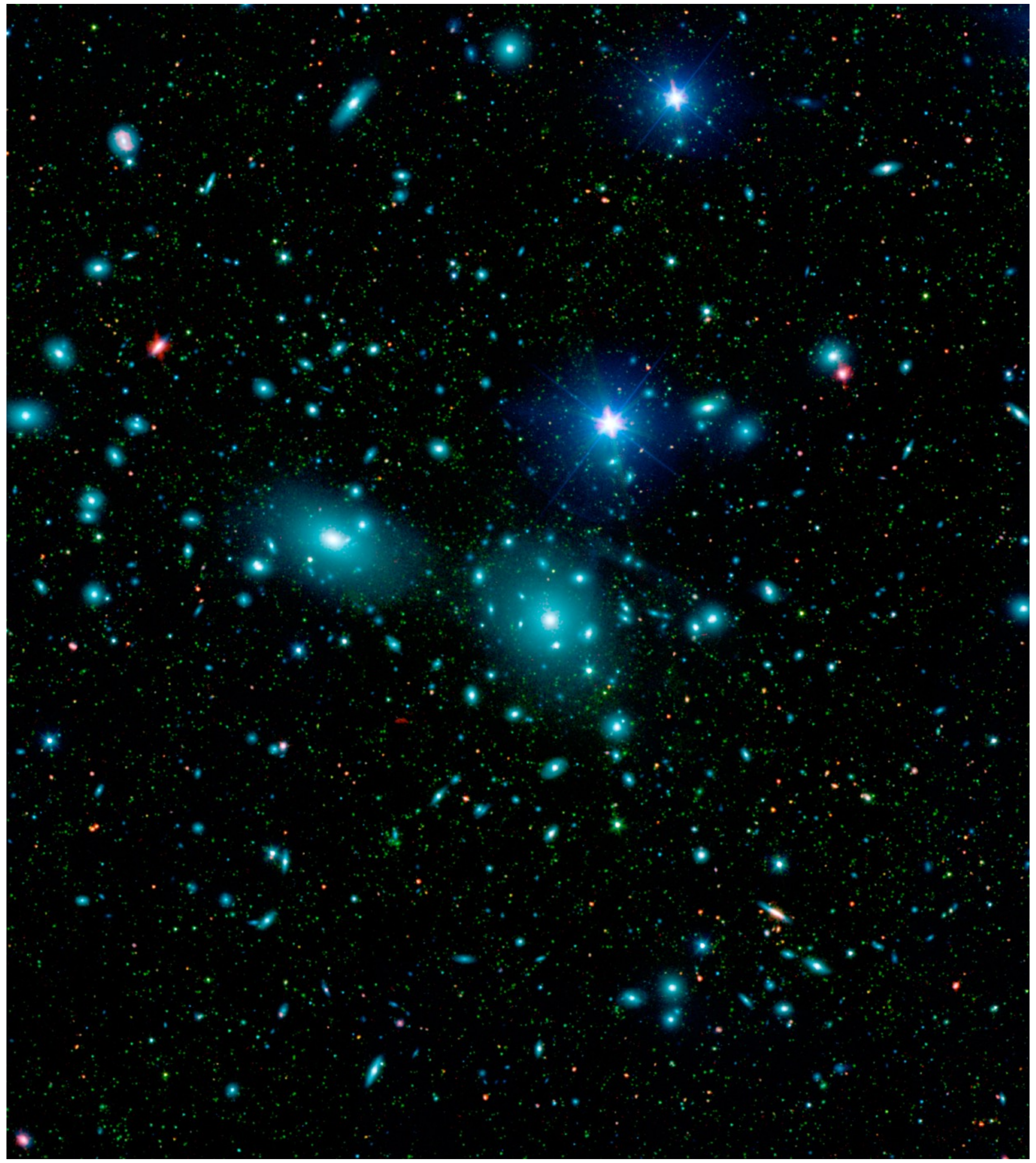
10.10. Standard-model final states

Possible keyword choices for the `gSIM_FLUX_FLAG_FINALSTATE` variable:

Keyword	Description
`kGAMMA`	γ-rays
`kNEUTRINO`	neutrinos
`kANTIPROTON`	anti-protons, $\bar{p}$
`kPOSITRON`	positrons, e^+
`kELECTRON`	electrons, e^-

1st exercise (30 minutes):

Calculate the J-factor and gamma-ray spectrum from DM annihilation in the Coma cluster



1. First calculate the total J-factor number:
   A. Figure out which CLUMPY module is suited for that (N.B.: do the calculation in 1D, not skymap mode). Hint: Use http://clumpy.gitlab.io/CLUMPY/doc_modules.html
   B. Find the total halo properties of the Coma cluster in 1104.3530. Caveat: CLUMPY requires $\rho_s = \rho(r_s) = 0.25\ \rho_0$
   C. Figure out how to provide the Coma halo parameters to CLUMPY. (N.B.: Define it as `CLUSTER` object in CLUMPY)
   D. Do the calculation first without taking into account halo substructure. Repeat the calculation with adding substructure, and play around with substructure parameters `gDM_SUBS_X` and `gCLUSTER_SUBS_X`
2. Calculate the spectrum at Earth between 100 GeV and 10 TeV for a thermal annihilation cross section, m_{DM} = 10 TeV, and annihilation into tau leptons. Hint: Use the `–z` module.

1st exercise: Solution

- Solution 1A:

Code:

```
[~/MPP-Cloud/Science/Clumpy/Tutorial_ISAPP2021] [10:00:01] > clumpy -h

                    v3.1, git-ID a8263b3 (157 commits after last tag)

                              Simulation mode: h

 Incomplete or wrong command line input - mind the correct usage:
  [M(r)]                   ] clumpy -h0  -i param_file -any_parameter (or --any_parameter)
  [rho_sm+<sub>(r)         ] clumpy -h1  -i param_file -any_parameter (or --any_parameter)
  [Jsm+<sub>(alpha_int)    ] clumpy -h2  -i param_file -any_parameter (or --any_parameter)
  [Jsm+<sub>(theta)        ] clumpy -h3  -i param_file -any_parameter (or --any_parameter)
  [2D-skymap Jsm+<sub>(halo)] clumpy -h4  -i param_file -any_parameter (or --any_parameter)
  [2D-skymap Jsm+sub(halo) ] clumpy -h5  -i param_file -any_parameter (or --any_parameter)
  [alpha_int(f*Jtot)       ] clumpy -h6  -i param_file -any_parameter (or --any_parameter)
  [dist(f*Jpointlike)      ] clumpy -h7  -i param_file -any_parameter (or --any_parameter)
  [sigma_p(R)              ] clumpy -h8  -i param_file -any_parameter (or --any_parameter)
  [I(R)                    ] clumpy -h9  -i param_file -any_parameter (or --any_parameter)
  [beta_ani(r)             ] clumpy -h10 -i param_file -any_parameter (or --any_parameter)

N.B.: To print default parameters and write them to a file, use -D (or "-h1D", "-h2D", etc.)
      Append flag --all or -a to print/write ALL possible parameters readable by CLUMPY
N.B.: Default output is both displays on screen and printing to files.
        => use option -d for displays only (e.g., -h1 -d)
        => use option -p for print only (e.g., -h1 -p)
N.B.: The order of options does not matter, except for the very first option,
      which must be the simulation module (in this case -hX).
```

CLUMPY website:

7.2.4. —h2: *J*-factors (1D)

Calculates $J(\alpha_{\rm int})$ towards a the halo centers defined in `gLIST_HALOES`. Note that triaxial haloes can be studied as well.

```
$ clumpy -h2 -i clumpy_params_h2.txt
```

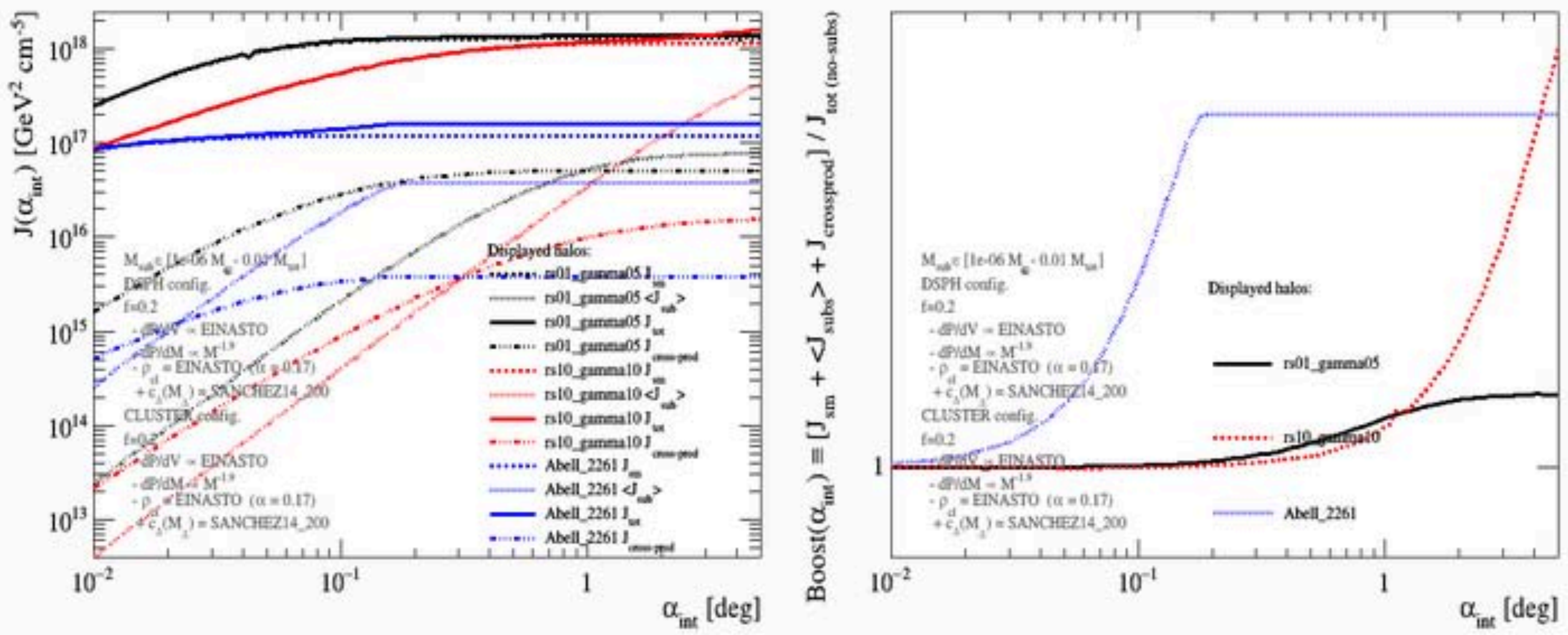


Fig. 7.16 J-factors (of smooth, sub-continuum and total components) of the haloes in `data/list_generic.txt` as a function of the integration radius $\alpha_{\rm int}$ of the search cone $\Delta\Omega$.

If enabled with the `gSIM_IS_WRITE_FLUXMAPS = True`, also the corresponding fluxes are calculated. The corresponding ROOT figure is shown in Fig. 6.29.

1st exercise: Solution

- Solution 1B:

Journal of Cosmology and Astroparticle Physics
An IOP and SISSA journal

Dark matter searches with Cherenkov telescopes: nearby dwarf galaxies or local galaxy clusters?

Miguel A. Sánchez-Conde,[a,b,c] Mirco Cannoni,[d] Fabio Zandanel,[e] Mario E. Gómez[d] and Francisco Prada[e]

Cluster (1)	z (2)	D_L (3)	M_{200} (4)	R_{200} (5)	c_{200} (6)	r_s (7)	ρ_0 (8)	Best IACTs (9)	Observed? (10)
Perseus	0.0183	77.7	7.71	1.90	3.99	0.477	7.25	M,V	W,V,M
Coma	0.0232	101.2	13.84	2.30	3.78	0.609	6.44	M,V	H,V
Ophiuchus	0.0280	122.6	23.16	2.74	3.60	0.760	5.81	H	-
Virgo	0.0036	15.4	5.6	1.68	4.21	0.433	6.81	M,V,H	-
Fornax	0.0046	19.8	1.01	0.96	4.80	0.201	11.0	H	-
NGC5813	0.0064	27.5	0.27	0.62	5.42	0.115	14.5	M,V,H	-
NGC5846	0.0061	26.3	0.38	0.69	5.26	0.132	13.5	M,V,H	-

Table 6. Main physical parameters of the galaxy clusters selected for this study. *Columns:* (1)

Total Coma DM halo properties after 1104.3530:

Redshift $z = 0.0234$ (= distance 103 Mpc)

$R_\Delta = 2.3$ Mpc with $\Delta = 200$ times the critical density of the Universe

NFW profile with

- $r_s = 609$ kpc
- $\rho(r_s) = 1.61 \times 10^5\ M_{sun}/kpc^3$

1st exercise: Solution

- Solution 1C:

```
> $CLUMPY/bin/clumpy -h2 -D
```

Creates a suitable default parameter file `clumpy_params_h2.txt`

Inside the file, watch out for the line

```
#-------------------------------------------------------------------------
# External lists with specified objects to add to simulations
#-------------------------------------------------------------------------
gLIST_HALOES                    [-]          $CLUMPY/data/list_generic.txt    <string or -1>    List of external halos definitions
```

Search for that default file in the CLUMPY directory & modify it (or use as template for new file):

```
#****************************************************************************************************************#
#              [OBJECT LOCATION AND SIZE]           |            DM DISTRIBUTION (RHO_TOT)                  [TRIAXIALITY]                 #
# Name            Type       l       b      d     z    Rdelta  |      rhos       rs     prof.   #1    #2    #3      IsTriaxial  a   b   c   rot1  rot2  rot3  #
#  -               -       [deg]   [deg]  [kpc]   -    [kpc]   |  [Msol/kpc3]  [kpc]   [enum]   -     -     -          -       -   -   -   [deg] [deg] [deg] #
#****************************************************************************************************************#

Coma_ACO1656     CLUSTER  58.0791 87.9577  -1   0.0234  2.3e3        1.61e5      609.    kZHAO   1.0 3.0   1.0  # after Sanchez-Conde et al. (2011)
```

Sky coordinates matter for next exercise

1^{st} exercise: Solution

- Solution 1D:

To switch off substructures, modify in the parameter file `clumpy_params_h2.txt`:

```
gDM_SUBS_NUMBEROFLEVELS          [-]          0          <integer>          Number of multilevel substructures
```

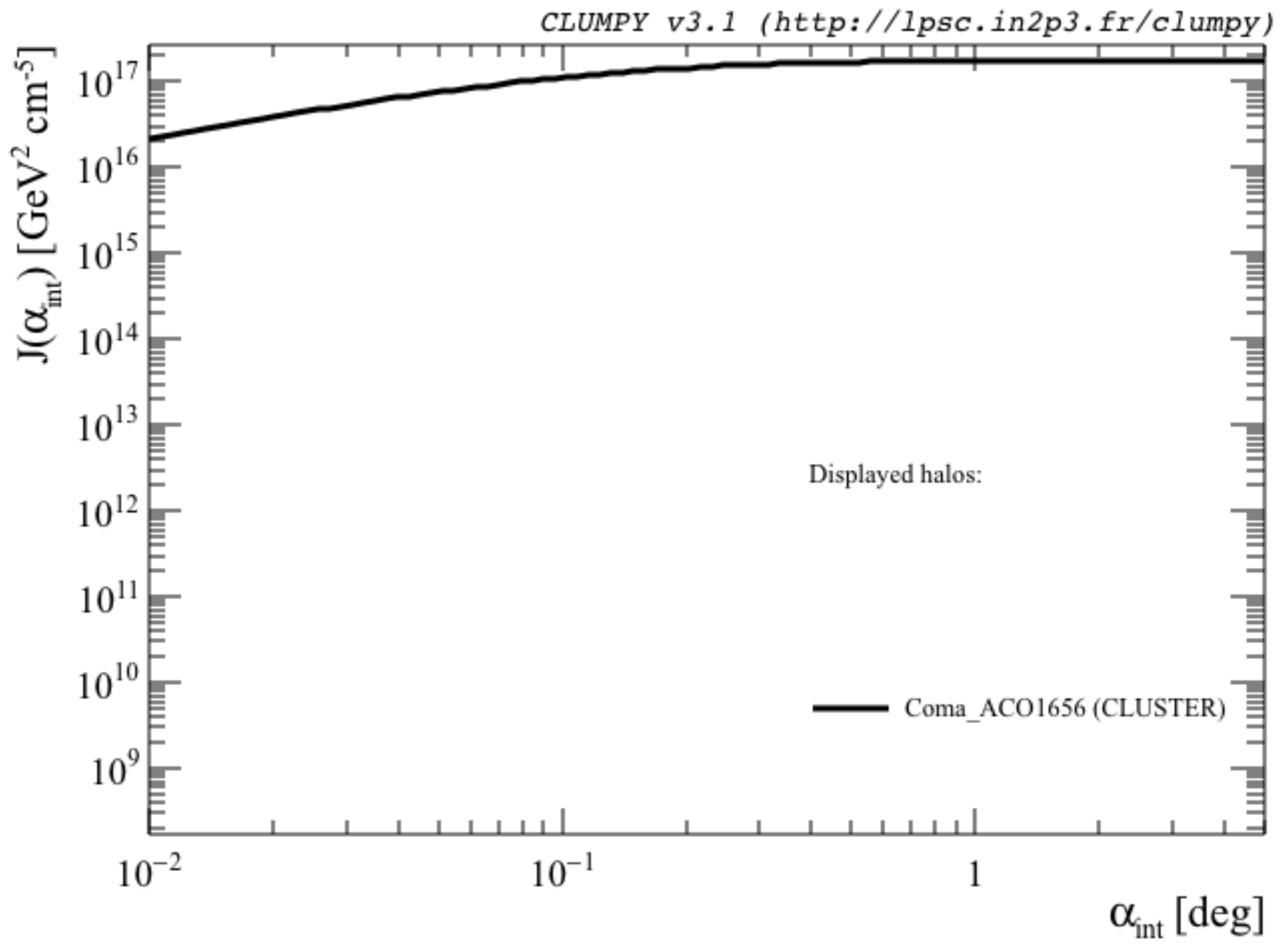
Alternatively, same result by setting

```
gCLUSTER_SUBS_MASSFRACTION          [-]          0.          <float>          Fraction of host halo mass bound in subhalos
```

Now run:

```
> $CLUMPY/bin/clumpy -h2 -i clumpy_params_h2.txt
```

Which creates the output plot (If ROOT is linked):



The total *J*-factor is

$J_{\text{no-subs}} = 1.75 \times 10^{17}\ \text{GeV}^2\ \text{cm}^{-5}$

N.B.: the angular extension of Coma is $1.2°$

1^{st} exercise: Solution

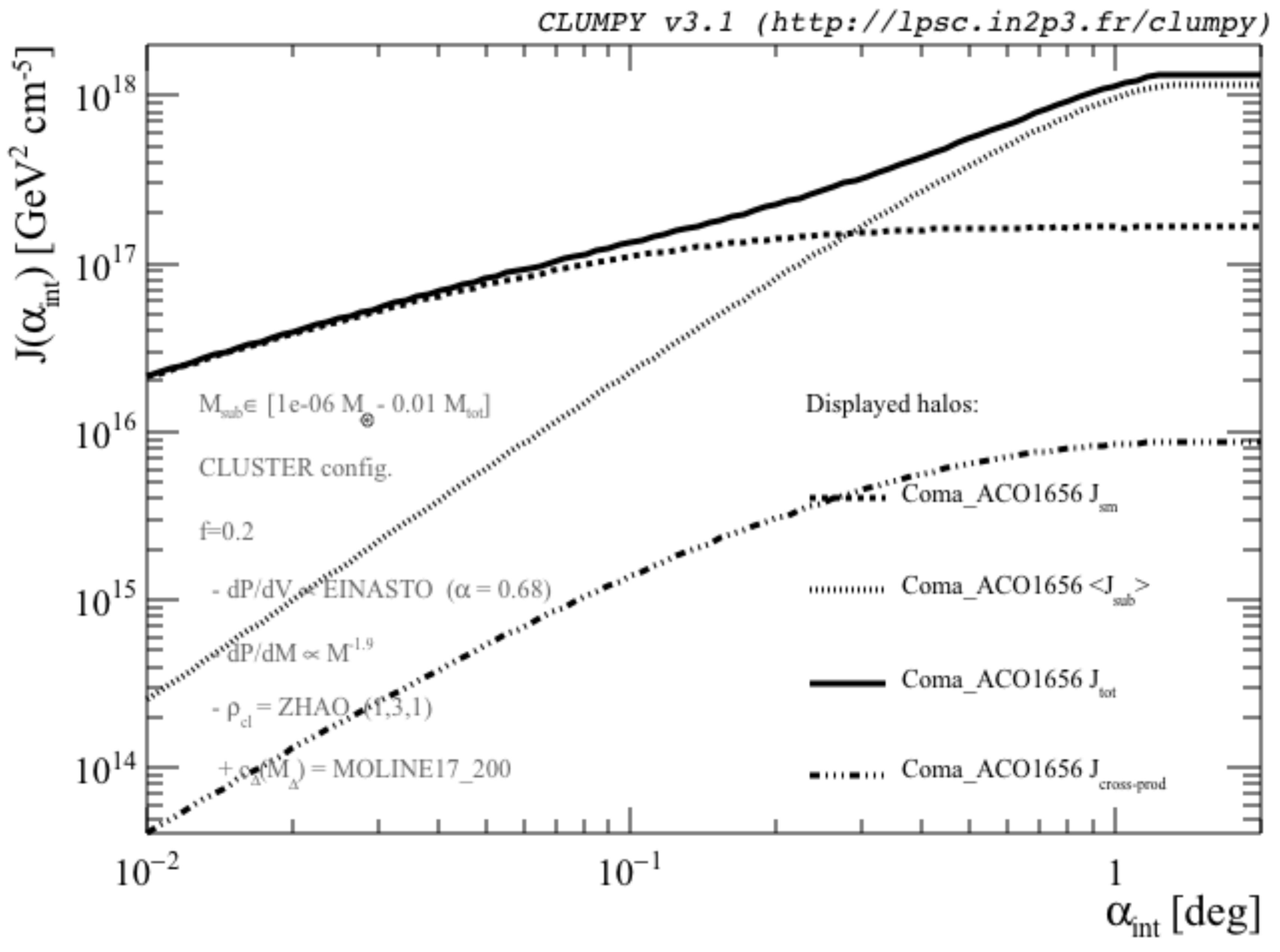
- Solution 1E:

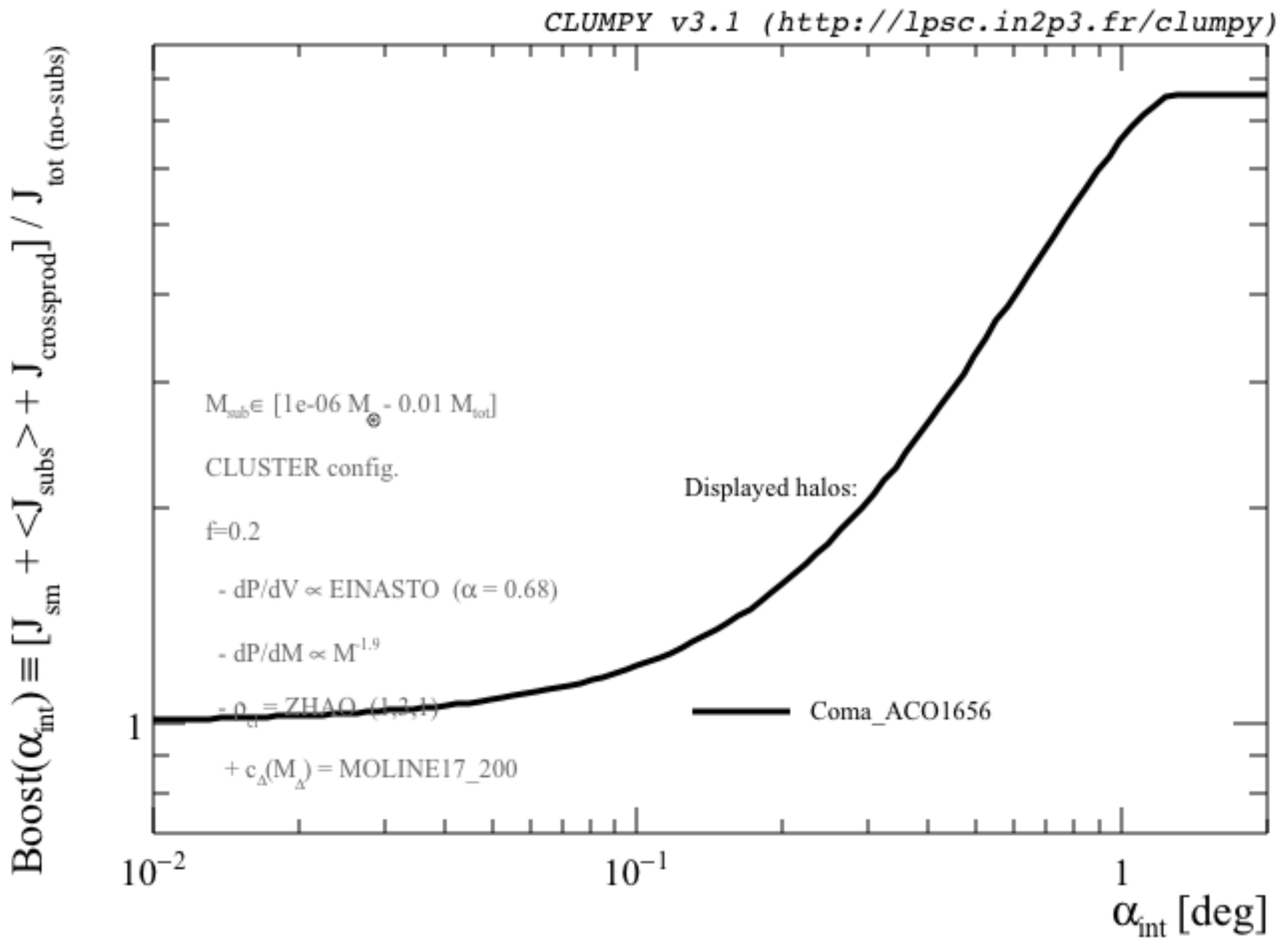
To switch on one level of substructures, modify in the parameter file `clumpy_params_h2.txt:`

```
gDM_SUBS_NUMBEROFLEVELS        [-]        1        <integer>        Number of multilevel substructures
```

And put e.g. 20% of cluster mass contained in substructures:

```
gCLUSTER_SUBS_MASSFRACTION        [-]        0.2        <float>        Fraction of host halo mass bound in subhalos
```





For this configuration, the total *J*-factor is

$J\text{tot} =$

1.32×10^{18} GeV² cm⁻⁵

(boosted by almost factor 8 compared to no substructures)

1st exercise: Solution

- Solution 2:

We use the `-z` module to compute the spectrum and create first a default parameter file:

```
> $CLUMPY/bin/clumpy -z -D
```

To answer the question, we need to modify then in the parameter file

```
gPP_BR                 [-]          0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,    <comma-sep list>  Branching channel ratio
```

```
gPP_DM_MASS_GEV        [GeV]        10000        <float>        [GeV] dark matter particle mass
```

```
gSIM_JFACTOR           [variable]   1.32e+18     <double>       Input J/D-Factor for plotting flux spectrum
```

```
gSIM_REDSHIFT          [-]          0.0234       <float>        Redshift of object(s) in the calculation
```

Choose an appropriate energy range to plot:

```
gSIM_FLUX_EMIN_GEV     [GeV]        100          <float>        Lower limit for integrated flux & shown spectrum
gSIM_FLUX_EMAX_GEV     [GeV]        10000        <float>        Upper limit for integrated flux & shown spectrum
```

And add an absorption profile (because redshift is not negligible):

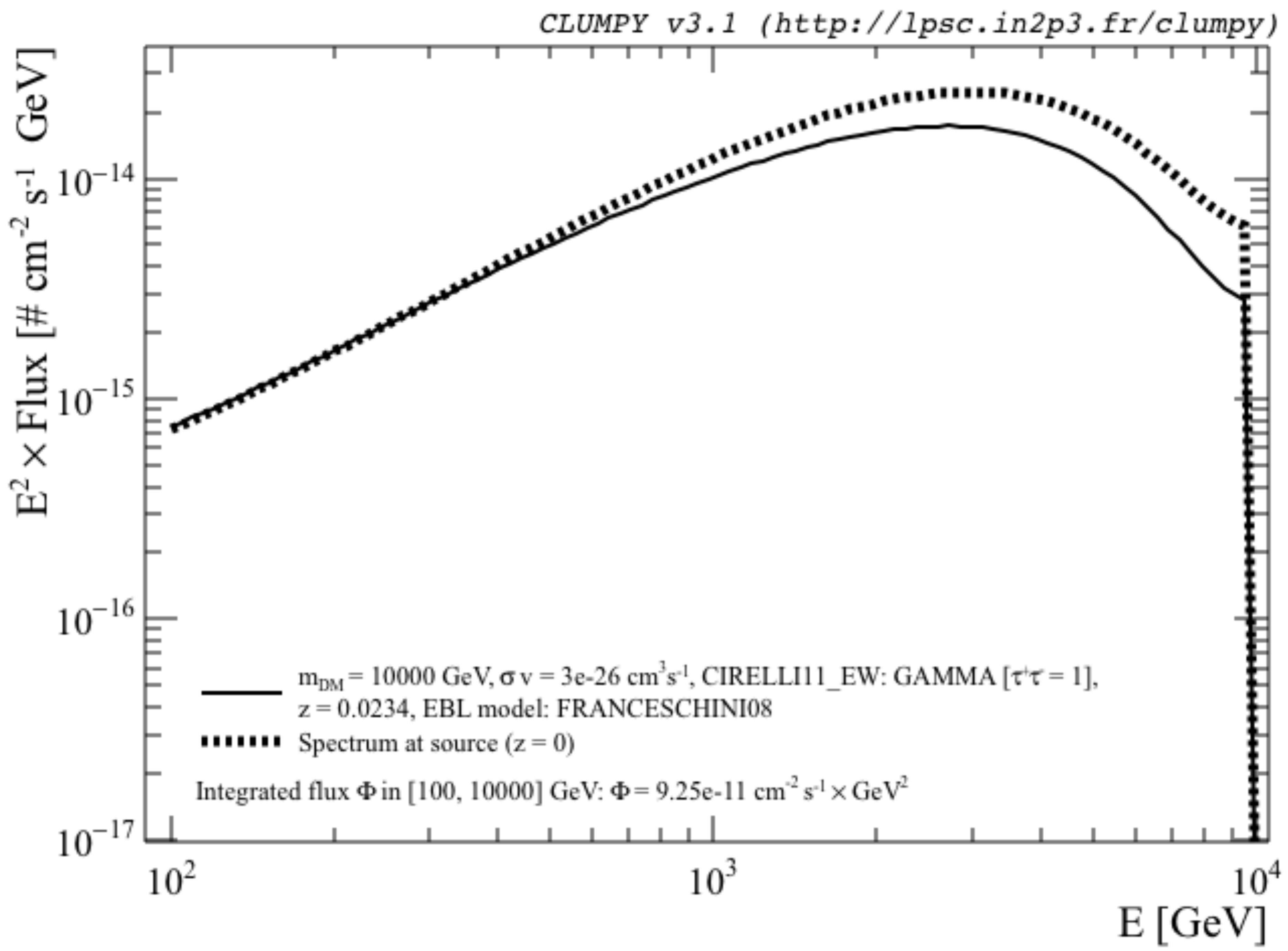
```
gEXTRAGAL_FLAG_ABSORPTIONPROFILE    [-]    kFRANCESCHINI08    <string>    EBL absorption profile
```

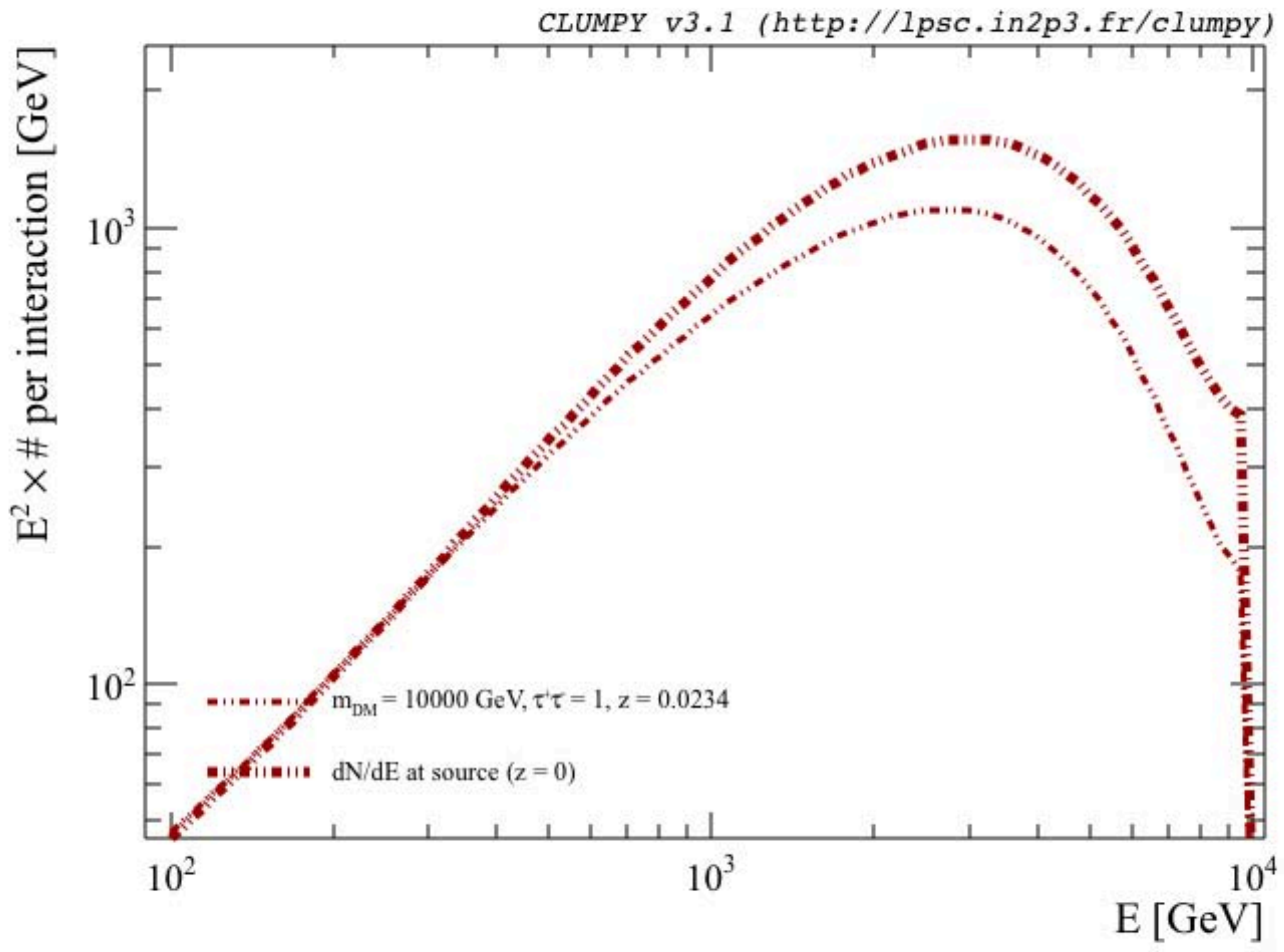
1st exercise: Solution

- Solution 2:

```
> $CLUMPY/bin/clumpy -z -i clumpy_params_z.txt
```

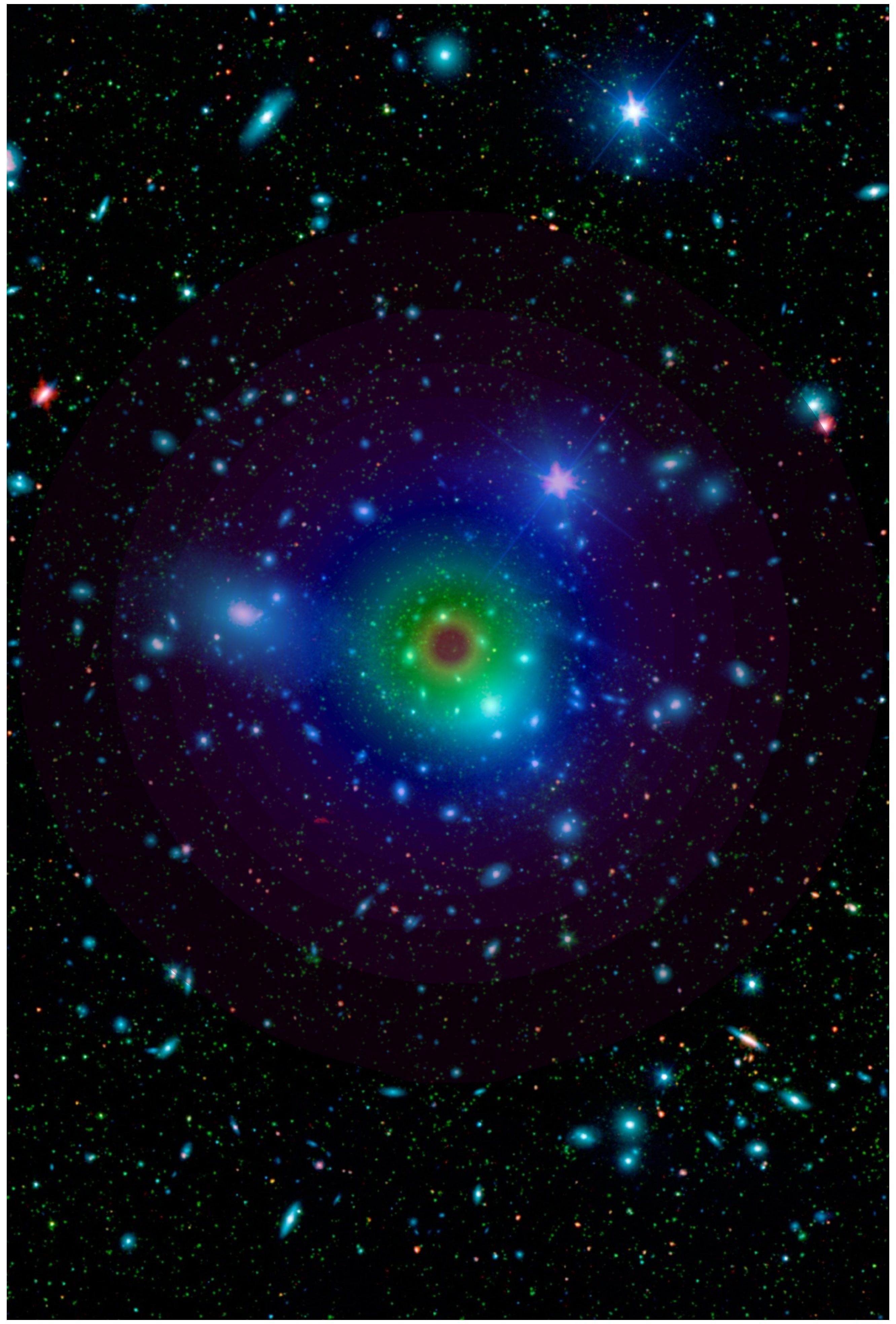
And we obtain the ROOT figures:





2nd exercise (30 minutes): ROOT is needed to be linked with Clumpy for this exercise

Create a 2D template map of Coma, overlay it with the Galactic halo J-factor skymap, and display all in Python

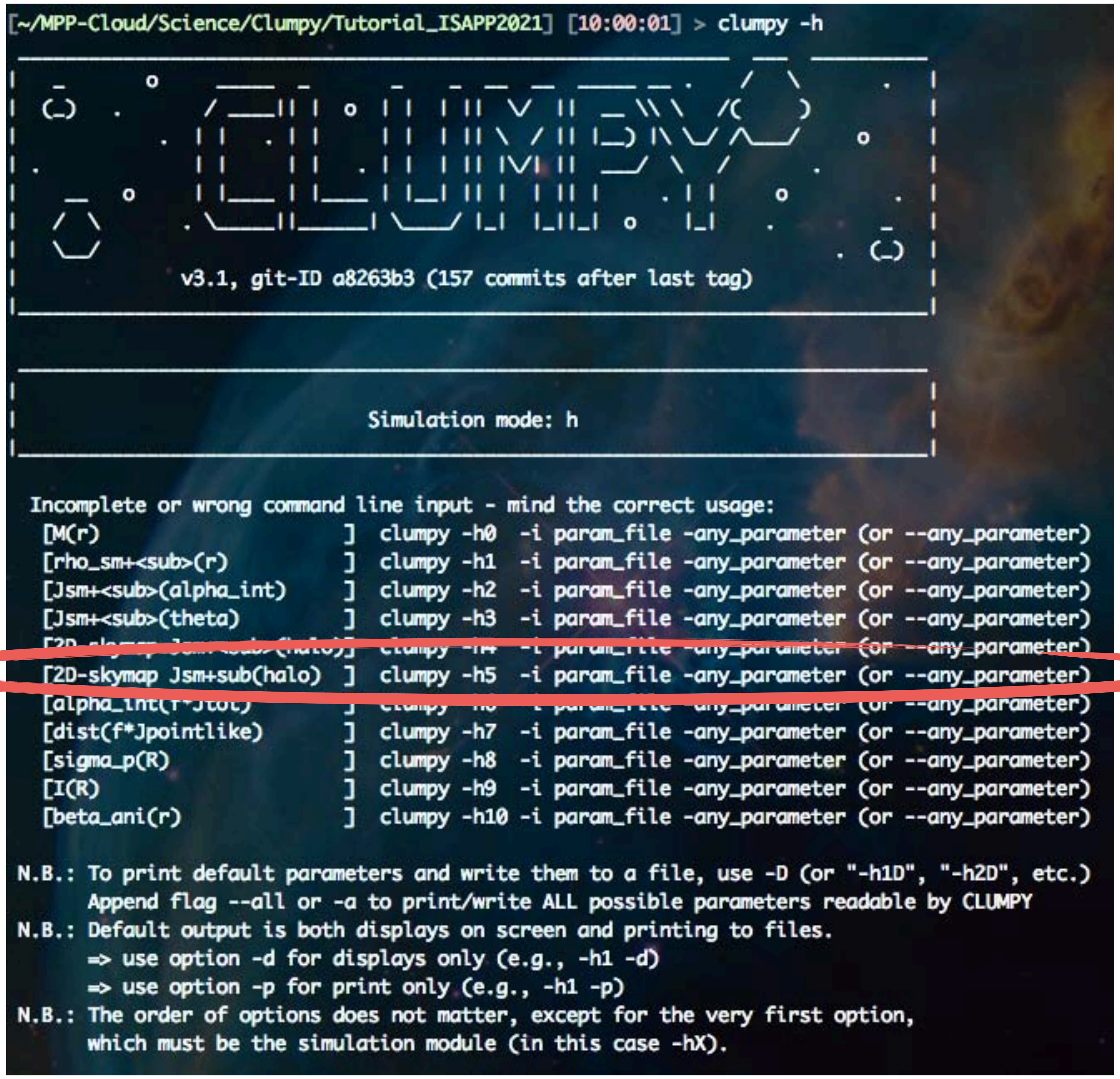


1. Create a 2D template of Coma with resolved substructures:
   A. Figure out which CLUMPY module is suited for that. Again, check http://clumpy.gitlab.io/CLUMPY/doc_modules.html. Create the default parameter file and adjust the parameters to load Coma, and such that the full cluster is in the field of view.
   B. Play with the `gSIM_USER_RSE` Relative-error threshold of resolved substructures, and again with the substructure properties `gCLUSTER_SUBS_X` to see in what variations they result.
   C. Investigate the fits output file with the CLUMPY -o module (http://clumpy.gitlab.io/CLUMPY/doc_module_o.html), Fitsviewer (fv), ds9 or Aladin.
2. Load the fits output into python with healpy and display the different components using healpy's plotting functions. Use the Jupyter notebook on http://clumpy.gitlab.io/CLUMPY/quick_start.html to start.
3. Bonus: Calculate a full sky map of the Galactic DM halo and also include in it the Coma cluster using the `-g6` module (need CLUMPY 3.1 released last Thursday)

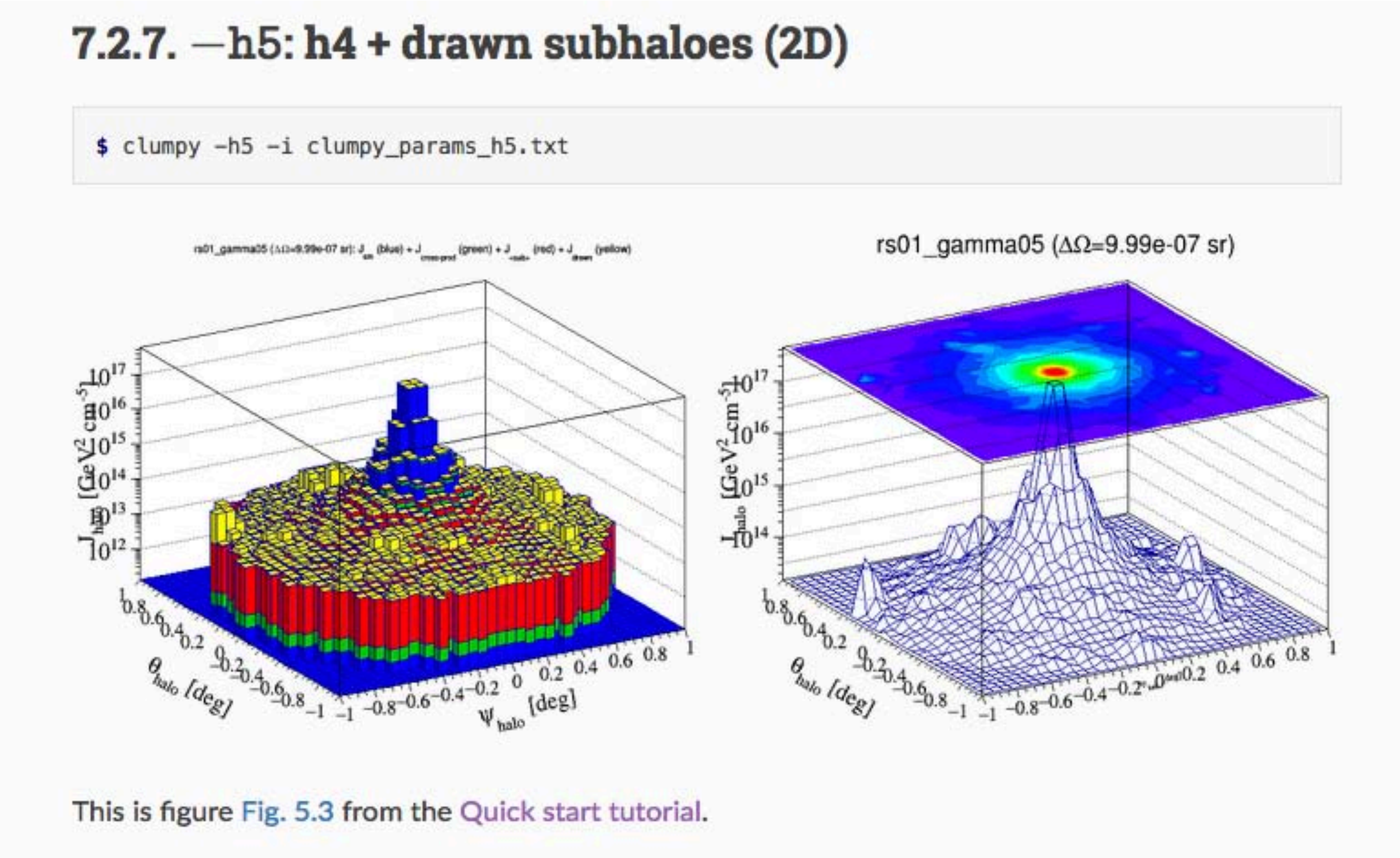
2nd exercise: Solution

- Solution 1A:

Code:



CLUMPY website:



2^{nd} exercise: Solution

- Solution 1A:

```
> $CLUMPY/bin/clumpy -h5 -D
```

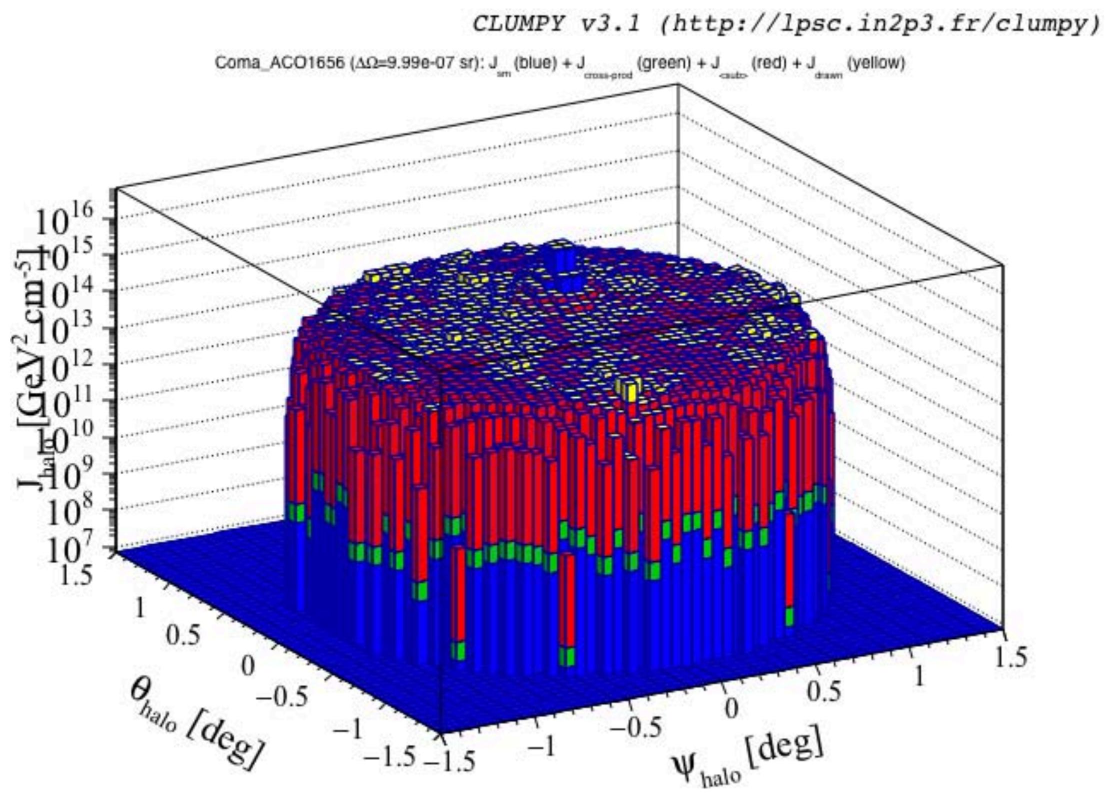
Then modify the parameter file `clumpy_params_h5.txt`:

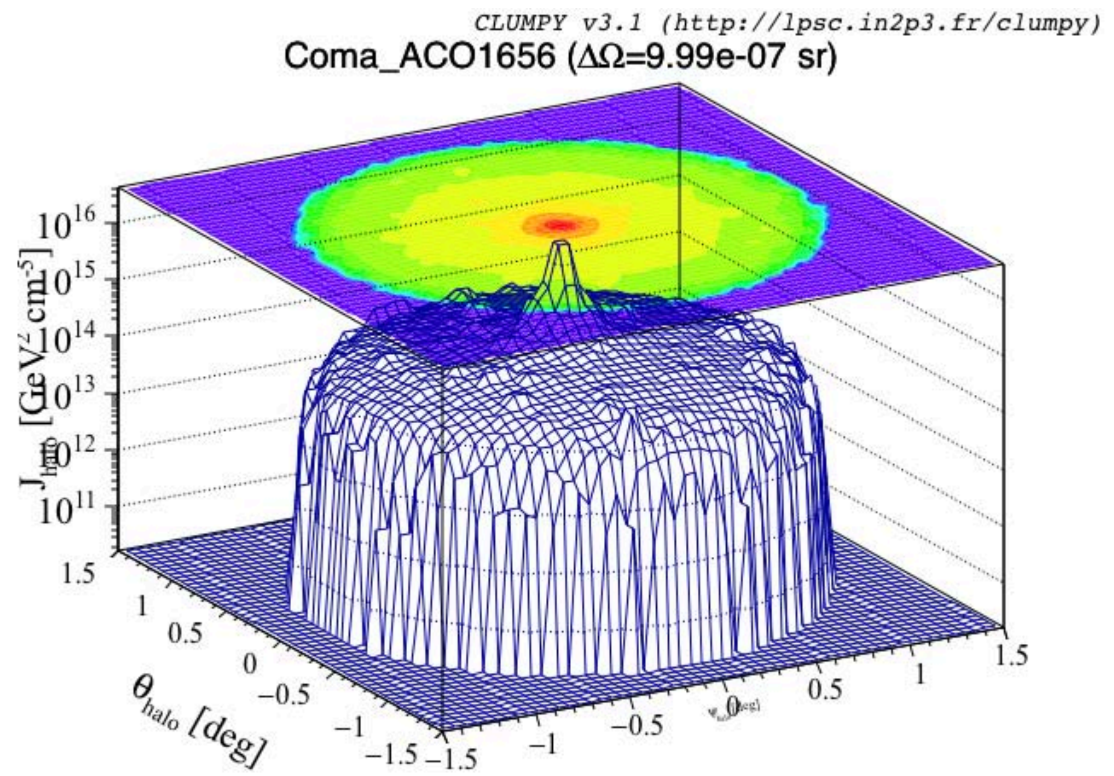
```
gLIST_HALOES           [-]      coma.txt        <string or -1>   List of external halos definitions
gLIST_HALONAME         [-]      Coma_ACO1656    <string>         Object selected out of a list of haloes
```

```
gSIM_THETA_SIZE_DEG    [deg]    3               <float>          Grid diameter in theta-dir.
```

```
> $CLUMPY/bin/clumpy -h5 -i clumpy_params_h5.txt
```

RE=10: 250 clumps resolved



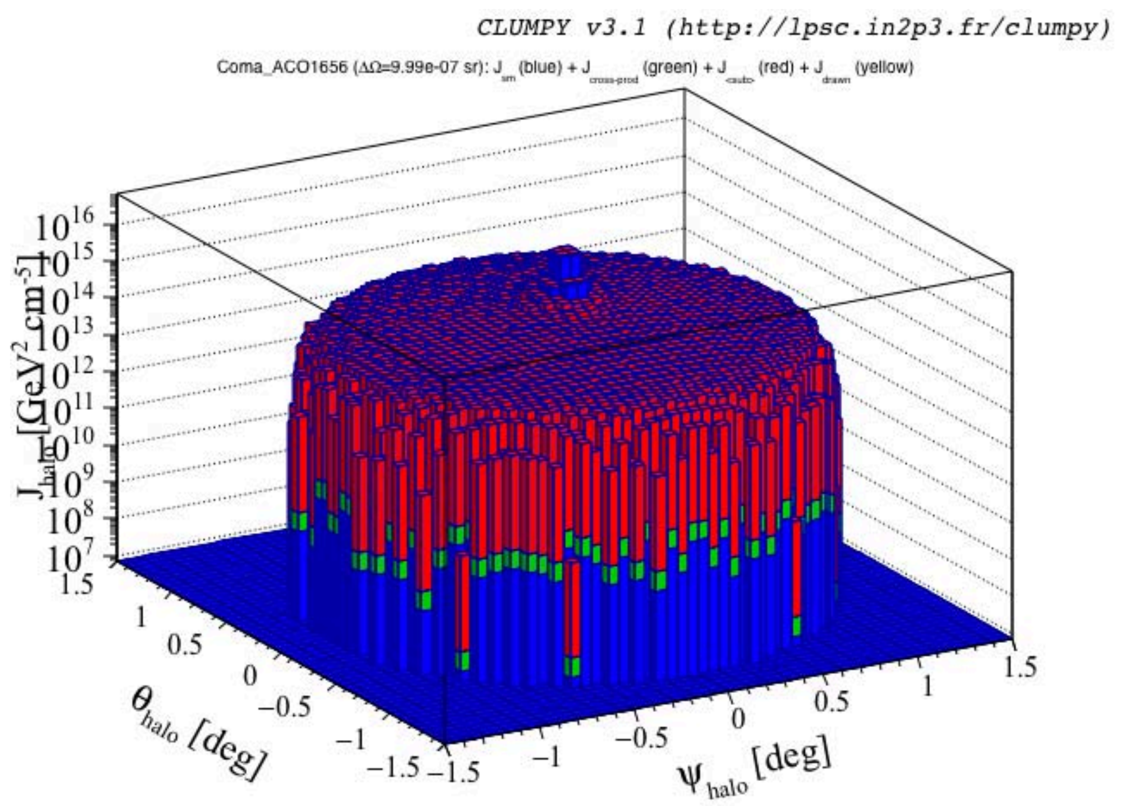


- Solution 1B:

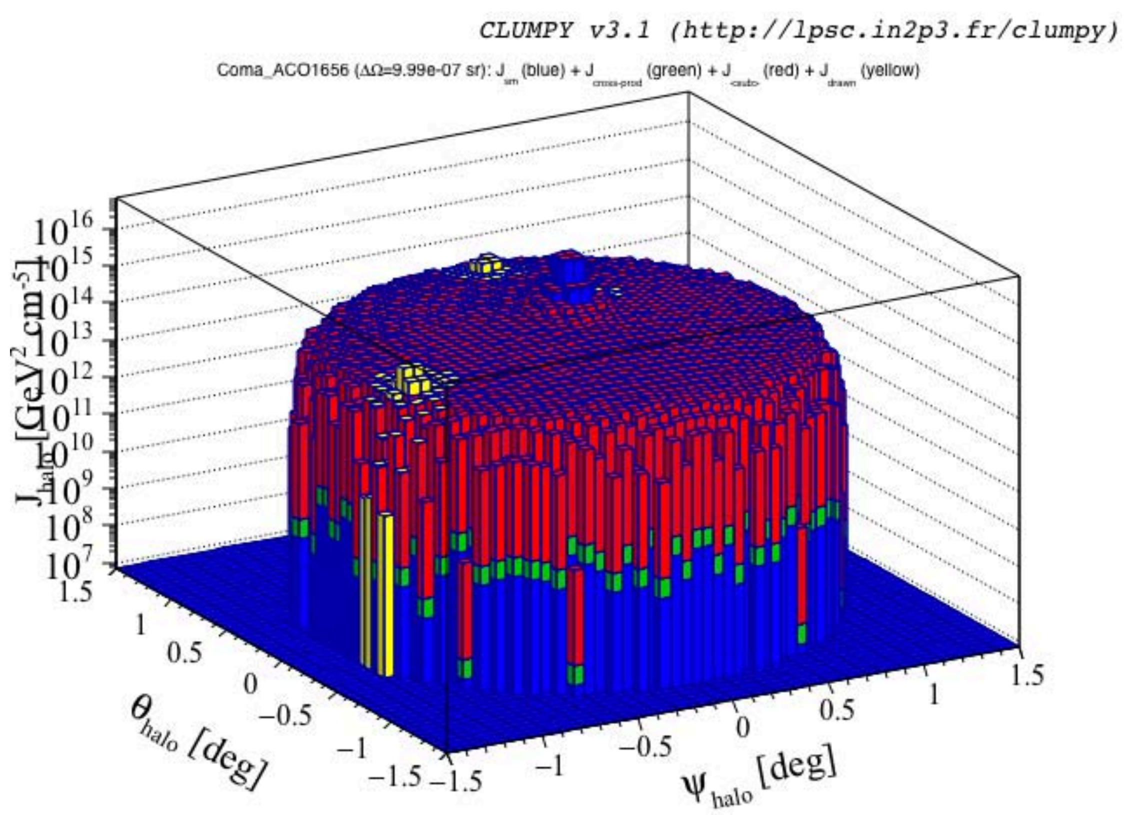
  Modifying:

```
gSIM_USER_RSE    [-]    10    <float>    Contrast (%) above which 2D subhalos are drawn
```

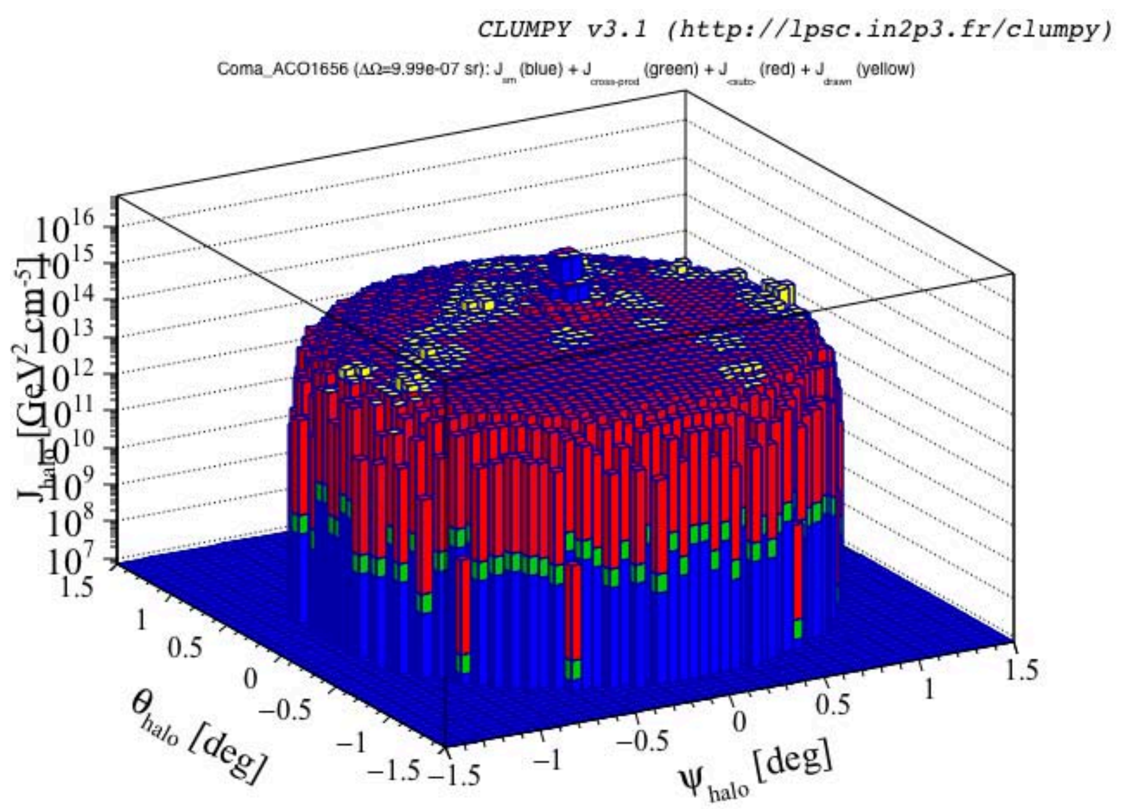
  gives:



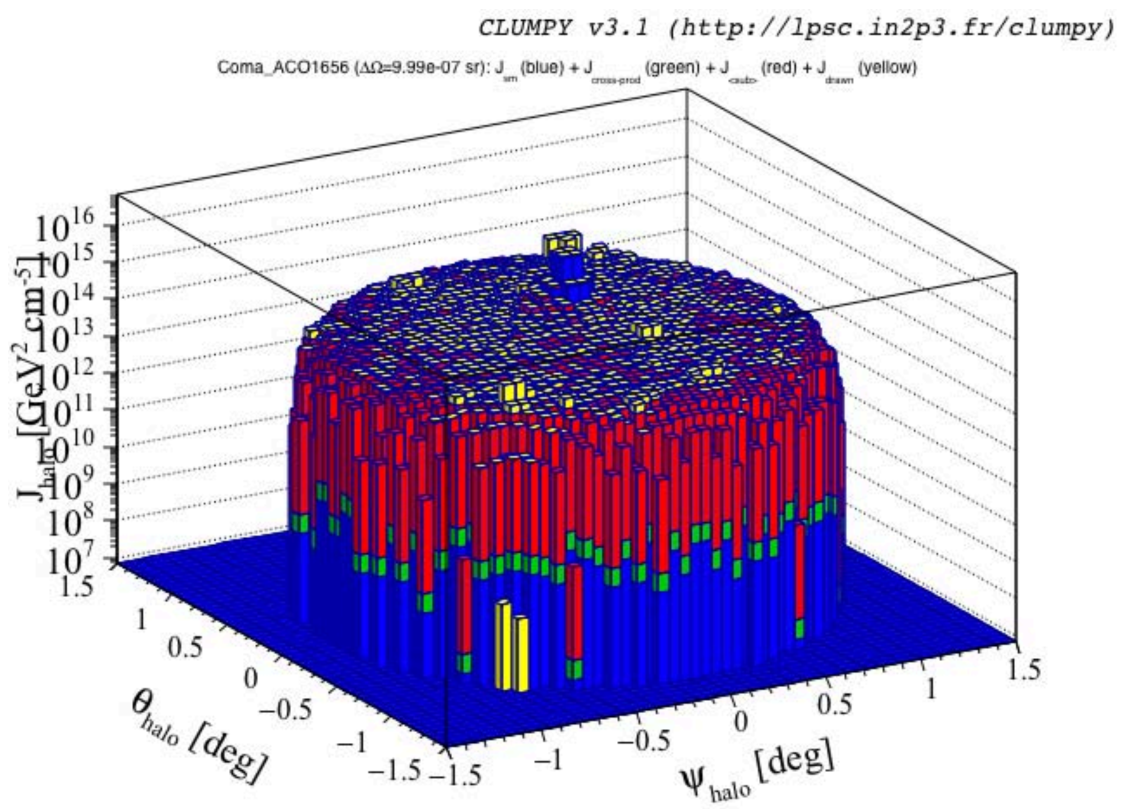
$\mathrm{RE}=100\%$: no clumps resolved



$\mathrm{RE}=50\%$: 3 clumps resolved



$\mathrm{RE}=25\%$: 28 clumps resolved

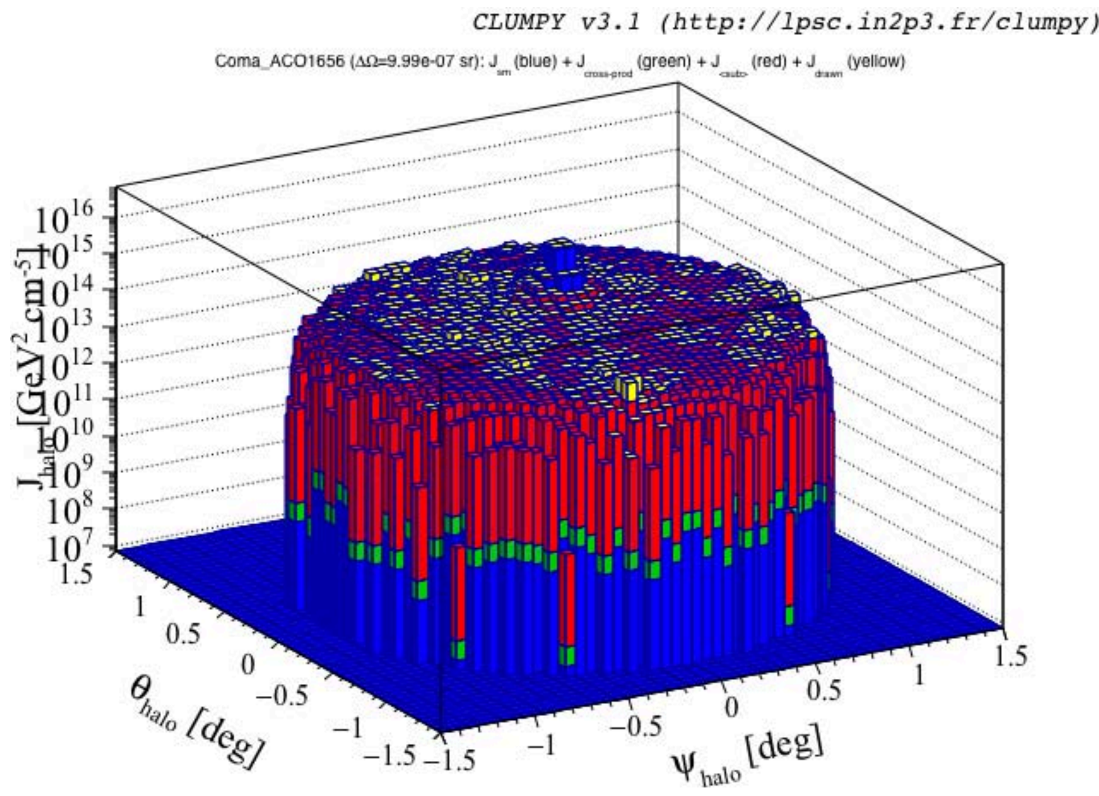


$\mathrm{RE}=5\%$: 1282 clumps resolved

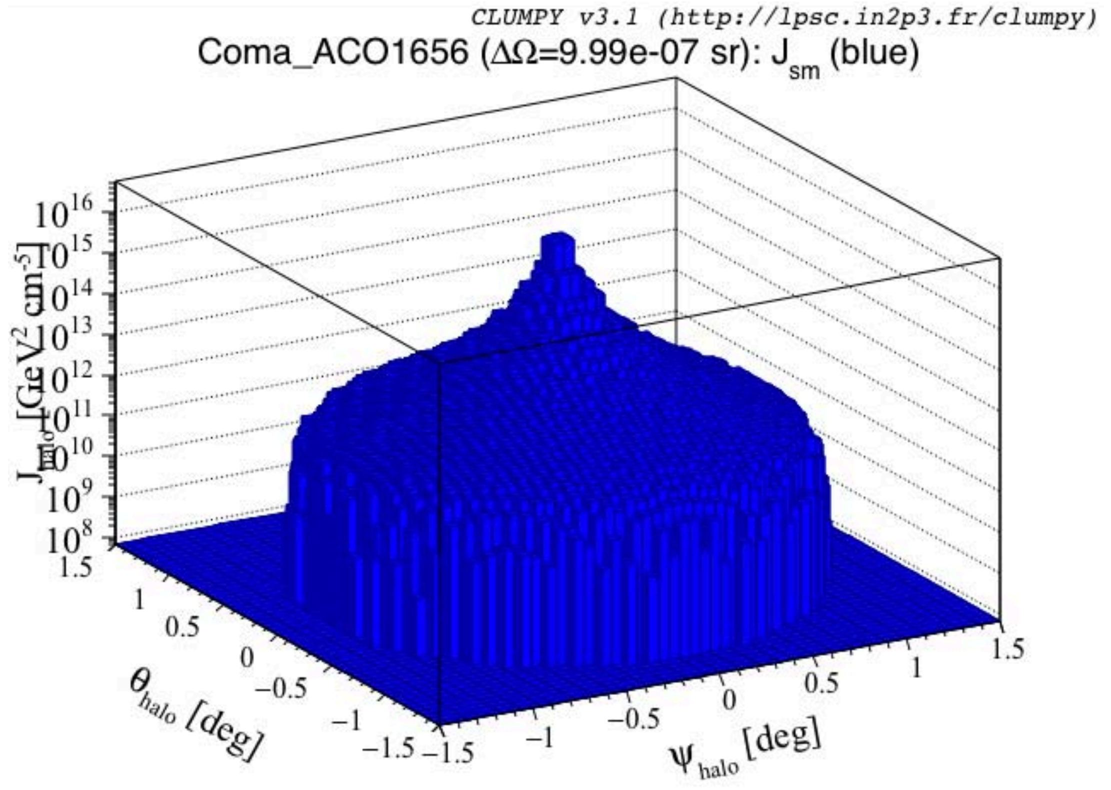
2^{nd} exercise: Solution

- Solution 1B:

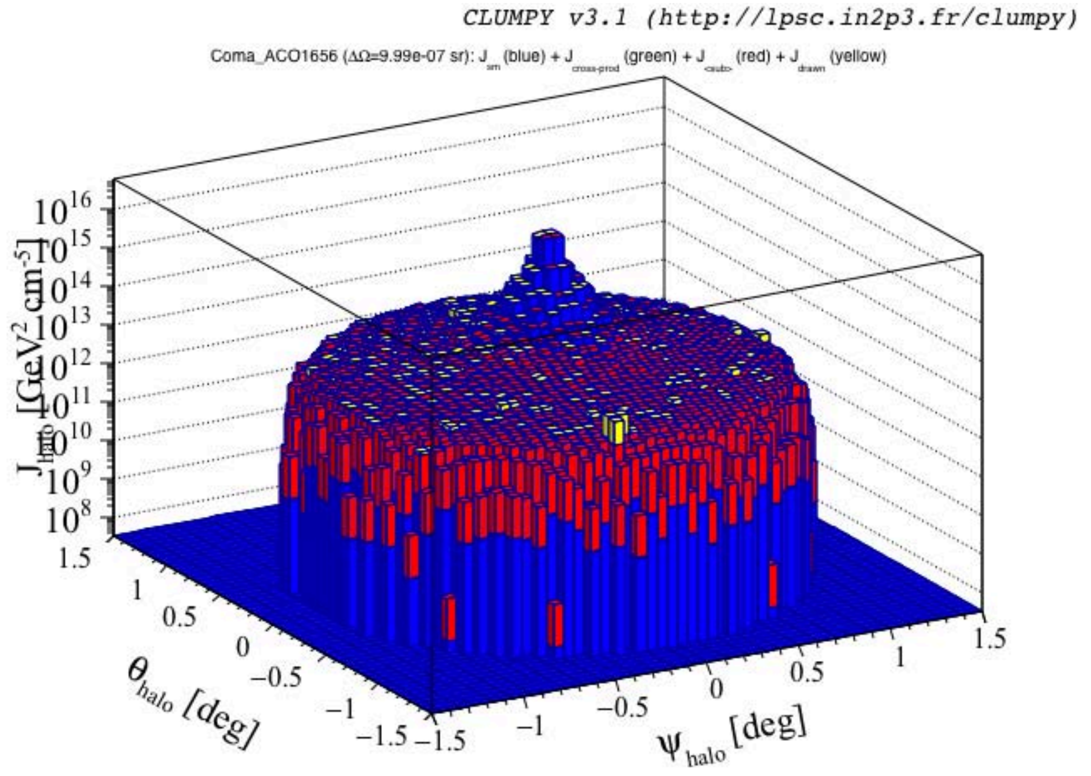
  Modifying substructure parameters gives:



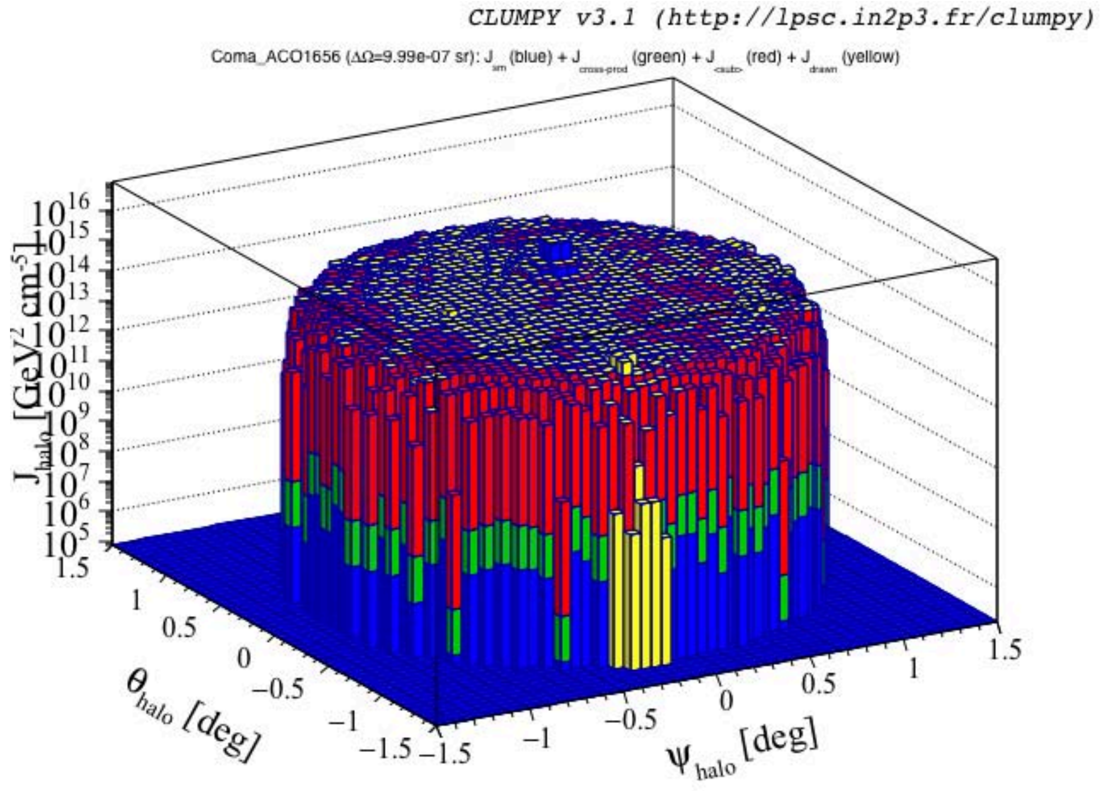
Default parameter file



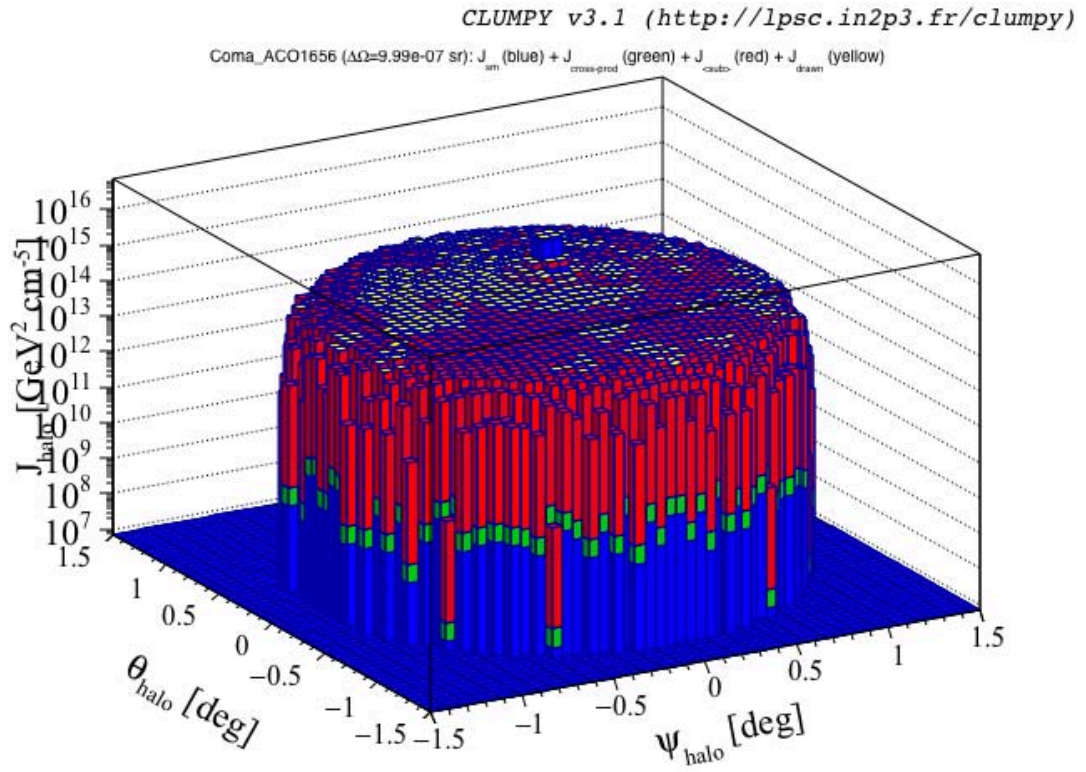
No substructure



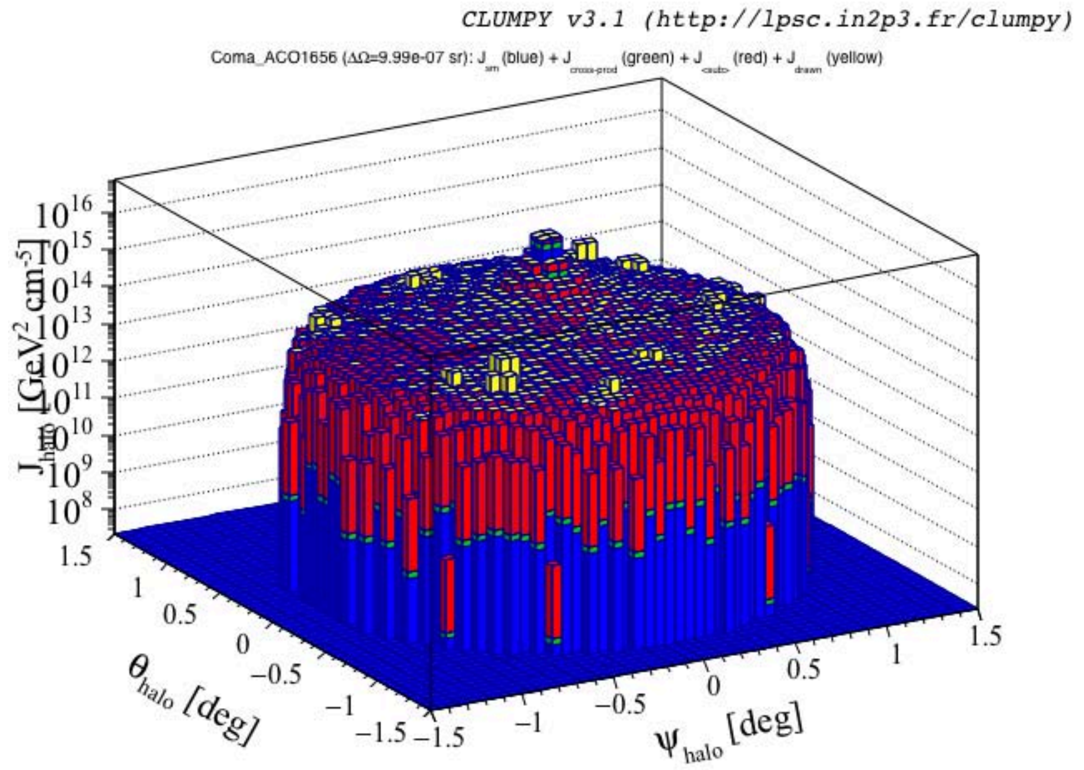
1% substructure



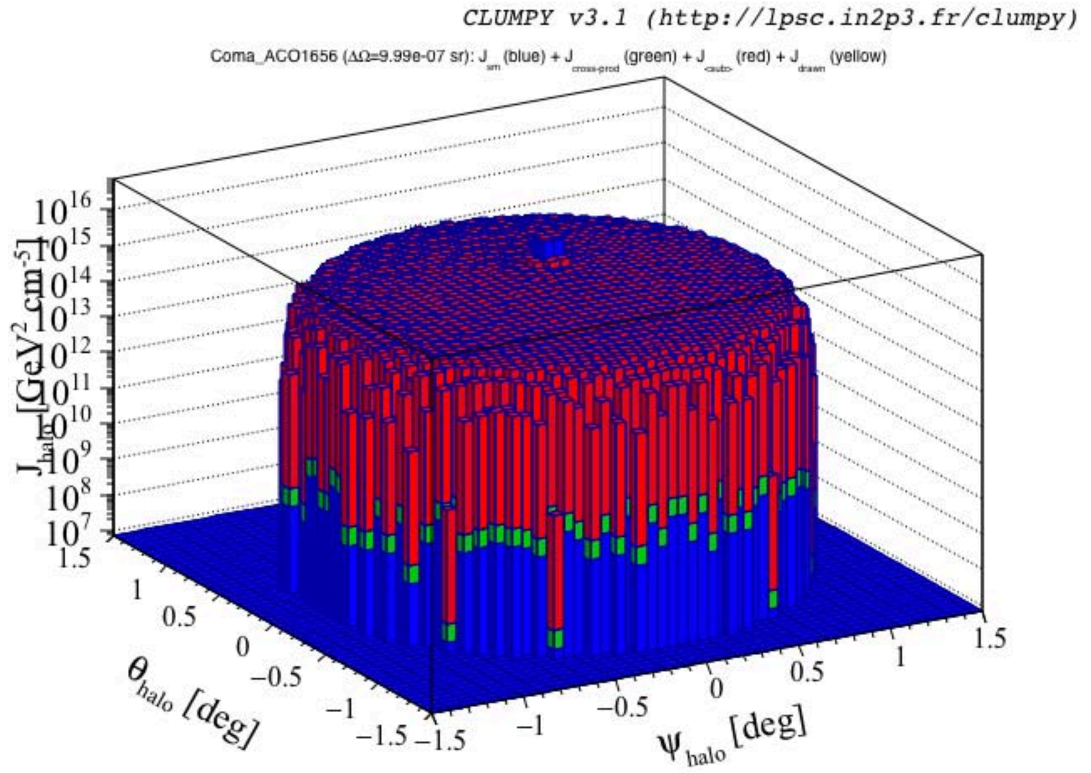
35% substructure



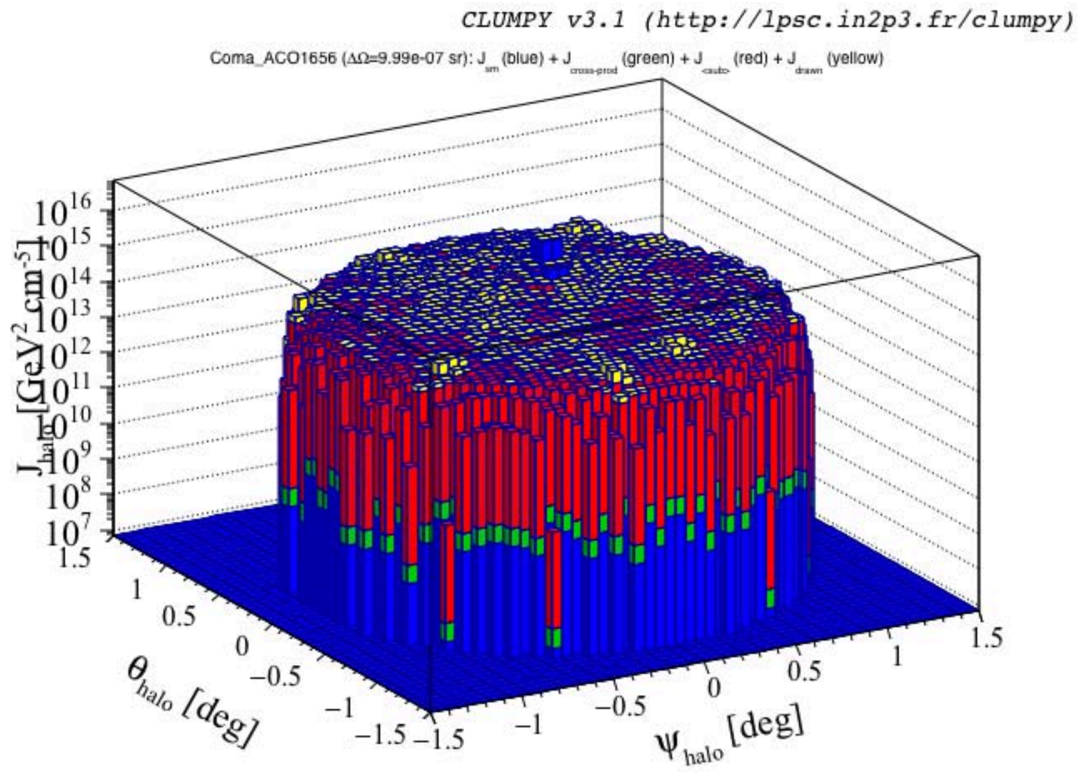
kMOLINE17_200 c(M,r)



Substructure traces the host



α_M=2



2 levels of substructure

2nd exercise: Solution

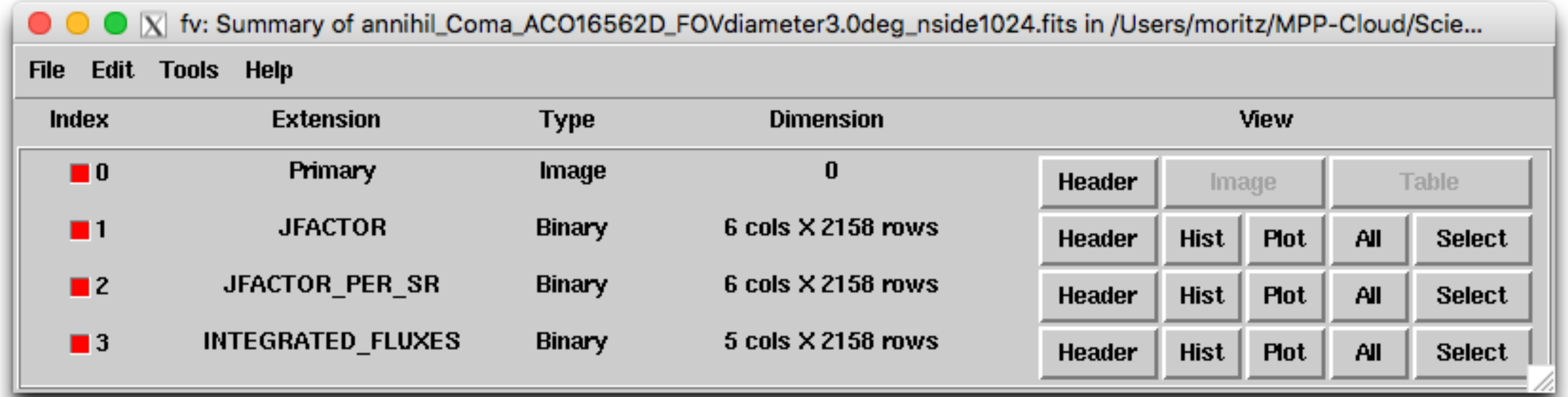
- Solution 1C:

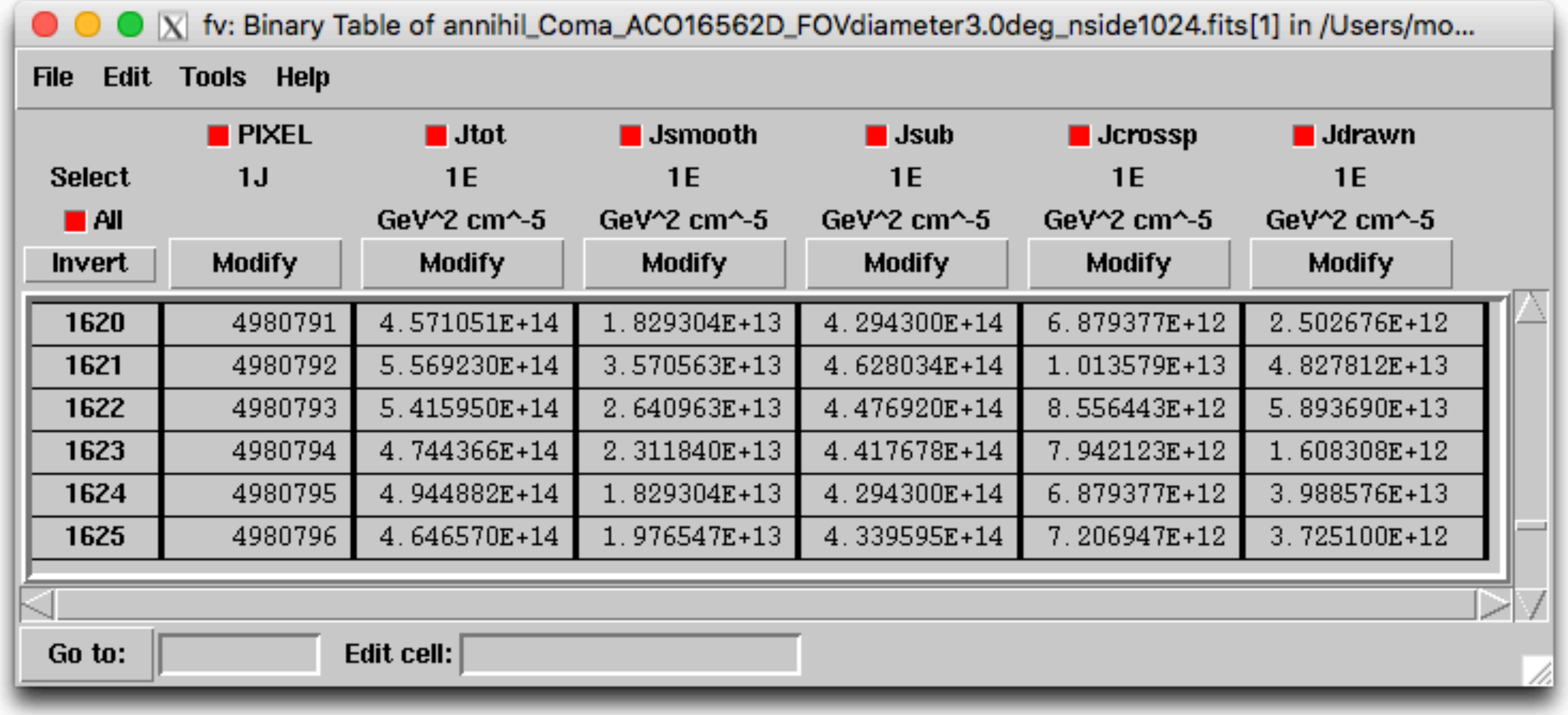
  See what is in the FITS file `output/annihil_Coma_ACO16562D_FOVdiameter3.0deg_nside1024.fits`:

```
> $CLUMPY/bin/clumpy -o2 -i output/annihil_Coma_ACO16562D_FOVdiameter3.0deg_nside1024.fits
```

```
>>>>> read FITS headers from file output/annihil_Coma_ACO16562D_FOVdiameter3.0deg_nside1024.fits ...
  => FITS file contains 3 extensions:

      Extension 1 :
        name of this extension: JFACTOR
        physical maps in this extension: 5
           map 0: PIXEL                [Healpix Pixel indices]
           map 1: Jtot                 Units: GeV^2 cm^-5
           map 2: Jsmooth              Units: GeV^2 cm^-5
           map 3: Jsub                 Units: GeV^2 cm^-5
           map 4: Jcrossp              Units: GeV^2 cm^-5
           map 5: Jdrawn               Units: GeV^2 cm^-5
        n_side of the maps: 1024
        ordering scheme of the maps: 1
        number of pixels (n_pix) in the maps: 2158

      Extension 2 :
        name of this extension: JFACTOR_PER_SR
        physical maps in this extension: 5
           map 0: PIXEL                [Healpix Pixel indices]
           map 1: Jtot_per_sr          Units: GeV^2 cm^-5 sr^-1
           map 2: Jsmooth_per_sr       Units: GeV^2 cm^-5 sr^-1
           map 3: Jsub_per_sr          Units: GeV^2 cm^-5 sr^-1
           map 4: Jcrossp_per_sr       Units: GeV^2 cm^-5 sr^-1
           map 5: Jdrawn_per_sr        Units: GeV^2 cm^-5 sr^-1
        n_side of the maps: 1024
        ordering scheme of the maps: 1
        number of pixels (n_pix) in the maps: 2158

      Extension 3 :
        name of this extension: INTEGRATED_FLUXES
        physical maps in this extension: 4
           map 0: PIXEL                [Healpix Pixel indices]
           map 1: Flux_gamma           Units: cm^-2 s^-1
           map 2: Flux_neutrino        Units: cm^-2 s^-1
           map 3: Intensity_gamma      Units: cm^-2 s^-1 sr^-1
           map 4: Intensity_neutrino   Units: cm^-2 s^-1 sr^-1
        n_side of the maps: 1024
        ordering scheme of the maps: 1
        number of pixels (n_pix) in the maps: 2158
```

Or just simply look at the file using fitsviewer (fv):

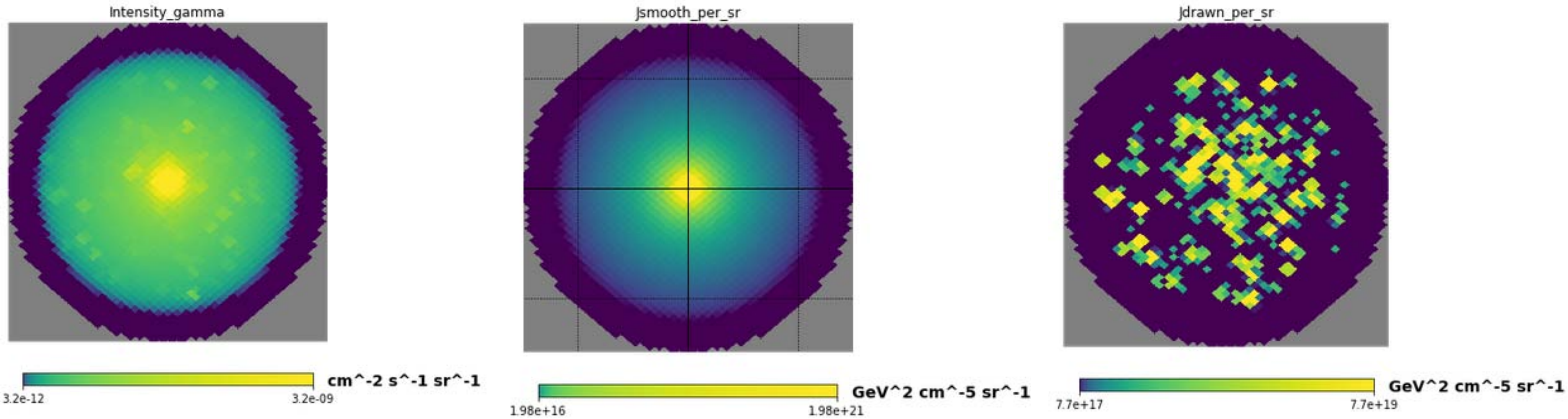




2nd exercise: Solution

- Solution 2:

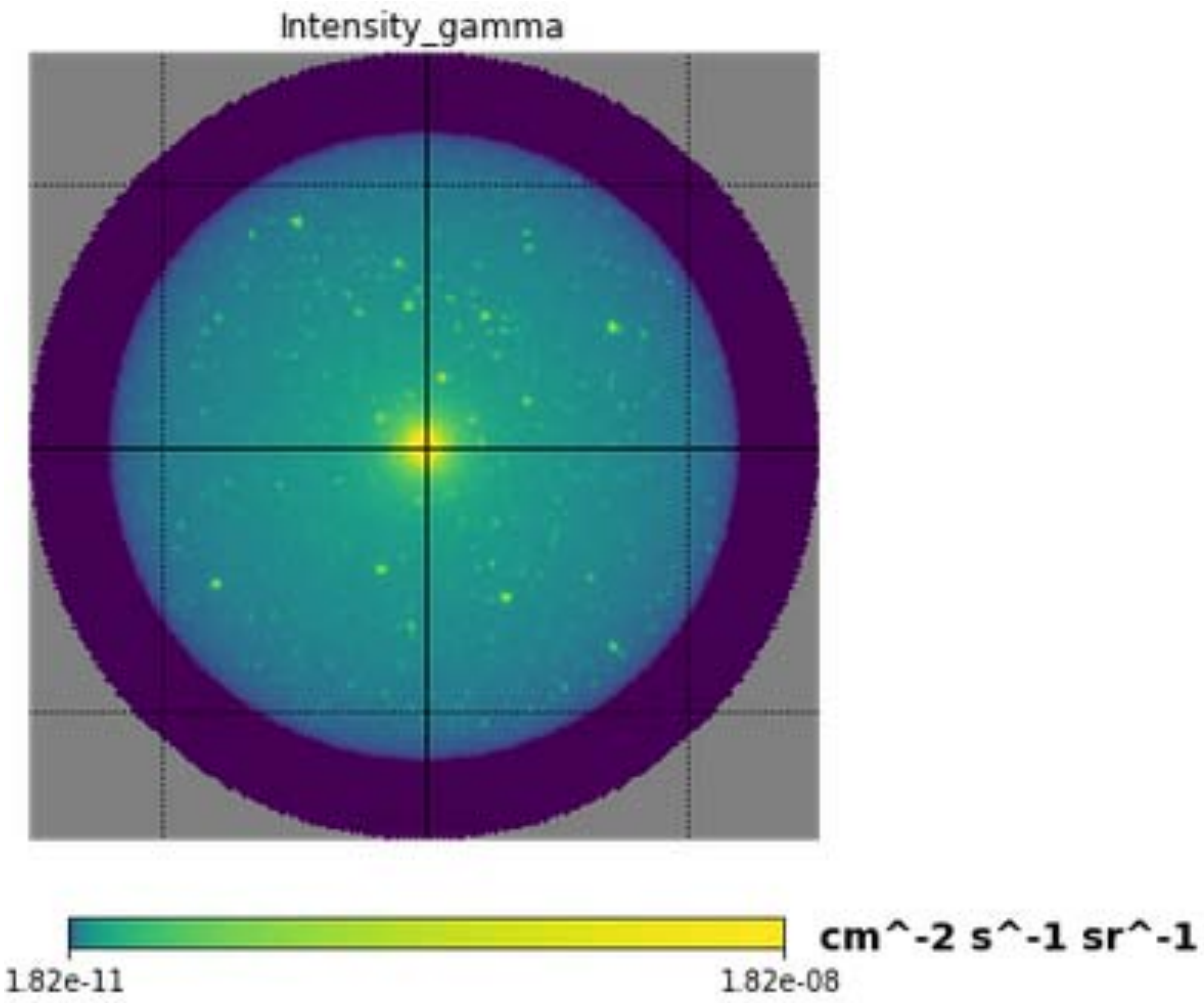
  See the Jupyter notebook for details:

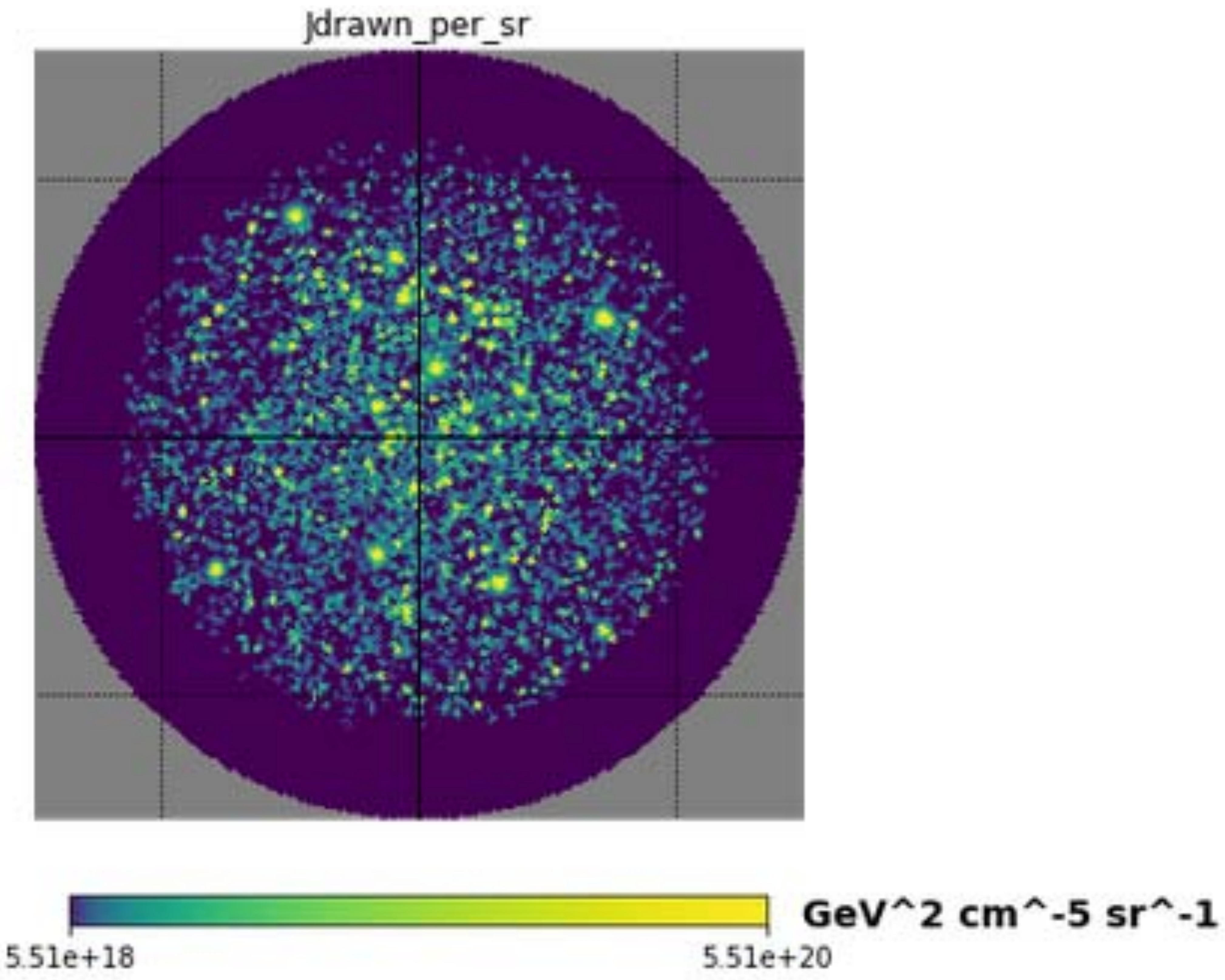


2nd exercise: Solution

- Solution 2 bonus:

  With fourfold resolution ($N\text{side} = 4096$):

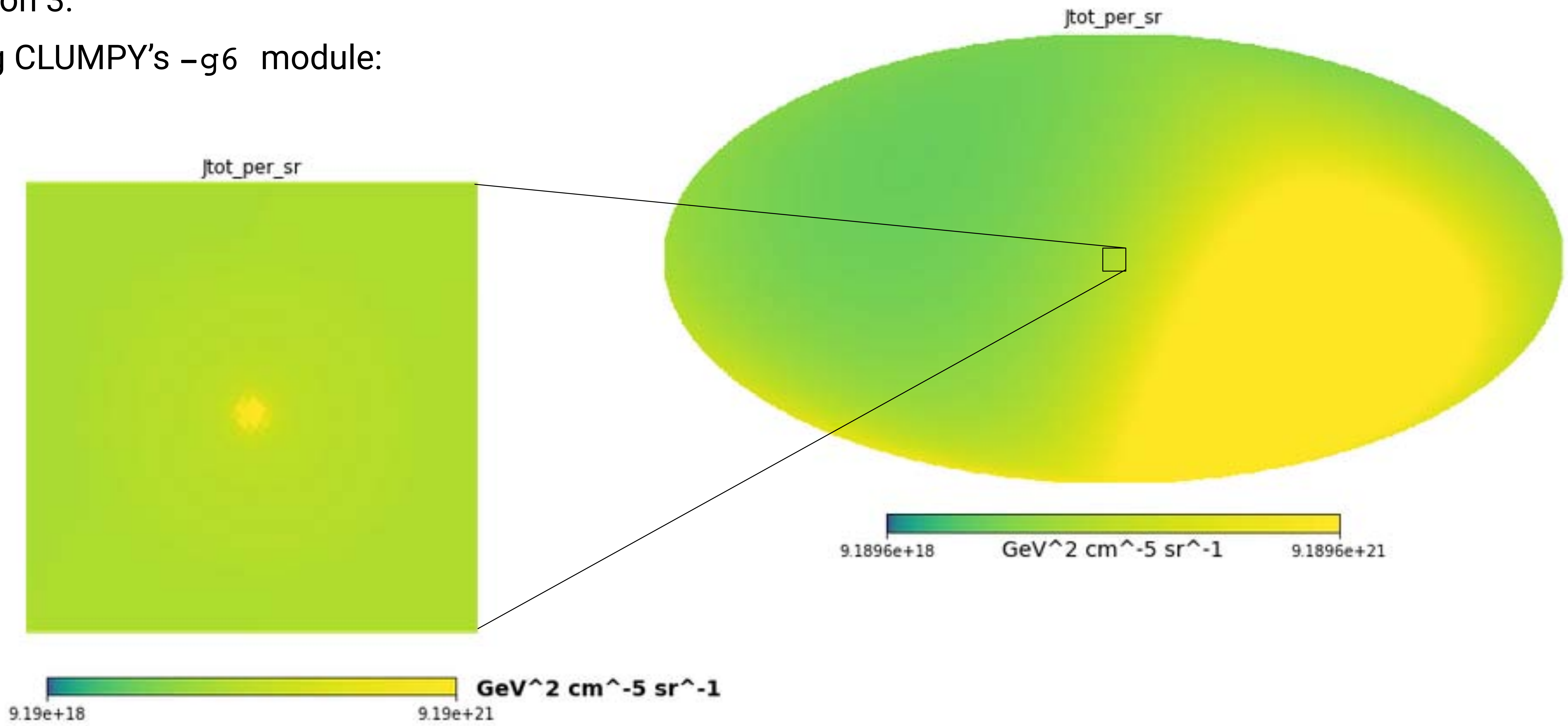




2nd exercise: Solution

- Solution 3:

  Using CLUMPY's –g6 module:



Close-out

- **Any general questions?**
- **If you face(d) problems installing or using CLUMPY, feel free (please!) to write to clumpy@lpsc.in2p3.fr**
- **In the course of this school: CLUMPY v3.1 is out!**
  - Tidal stripping model according to Stref et al. (2017) for Galactic subhalos
  - Implementation of distant-dependent mass-concentration models $\mathrm{c}(M,r)$
  - Arbitrary numeric (node) density profiles of halos as input possible

Thanks a lot for participating!