Extraction of physical quantities from numerical data cubes by use of the YT package

O. Stanchev [1], T. Veltchev [1], S. Donkov [2], B. Körtgen [3]

[1] *Faculty of Physics, University of Sofia; 5 James Bourchier Blvd., 1164 Sofia*
[2] *Department of Applied Physics, Technical University, 8 Kliment Ohridski Blvd., 1000 Sofia*
[3] *Universitat Hamburg, Hamburger Sternwarte, Gojenbergsweg 112, 21029 Hamburg, Germany*

Using the YT package in astronomy

YT (www.yt-project.org) is a flexible and cross-code software package for quantitative data analysis and visualization. It has been designed to be a common platform for simulation analysis, so that scripts can be shared across groups and analysis can be repeated by independent scientists. Historically, YT was initially developed to examine slices and projected regions through deeply nested adaptive mesh refinement (AMR) cosmological simulations conducted with ENZO (Bryan & Norman 1997; O'Sheaet al. 2004), but it was quickly repurposed to be a multi-code mechanism for data analysis and visualization. Primarily it was written in PYTHON but several core routines were written in C for fast computation. YT heavily utilizes the NUMPY library (Oliphant 2009) and is itself a PYTHON module suitable for direct scripting or access as a library.

Numerical data: simulation of cloud evolution

The data cube we use contains magnetohydrodynamical simulation of molecular cloud evolution including gravity. The cloud formation was modelled to occur through convergence of flows in warm neutral medium (initial temperature $T = 5\,000$ K, mean density 1 cm^{-3}), represented as two colliding, large-scale cylindrical streams. The latter are embedded in a numerical box of size (256 pc)^3 and each of them is 112 pc long, with a radius of 64 pc. They are given an initial supersonic inflow velocity (isothermal Mach number of 2) to collide at the centre of the numerical box. The subsequent cloud evolution was followed with a FLASH code (Fryxell et al. 2000) with maximal mesh refinement corresponding to scales ~ 0.125 pc. The initial turbulence with Mach number 0.4 decays and is not driven continuously. Turbulent motions at the considered late evolutionary stages are due to fluid motions, driven by gravity. Feedback by sink particles is not included in the simulation.

Source code for data projections: definitions

```
# add new derived field:
def _NumDensity(field, data):
    return (data["dens"]/(1.37*1.67e-24))

# project the data cube along arbitrary oriented axis:
def project(pf, angle, width = None, depth = None):

    x_end_point = pf.domain_right_edge[1] * tan((90.0-angle)*pi/180.0)
    y_end_point = pf.domain_right_edge[1]
    z_end_point = 0

    normal = [x_end_point, y_end_point, z_end_point]
    nv = [0, 0, pf.domain_right_edge[2]]

    p = OffAxisProjectionPlot(pf, normal, 'NumDensity', width=width,
    depth=(256.0, 'pc'), north_vector=nv)
```

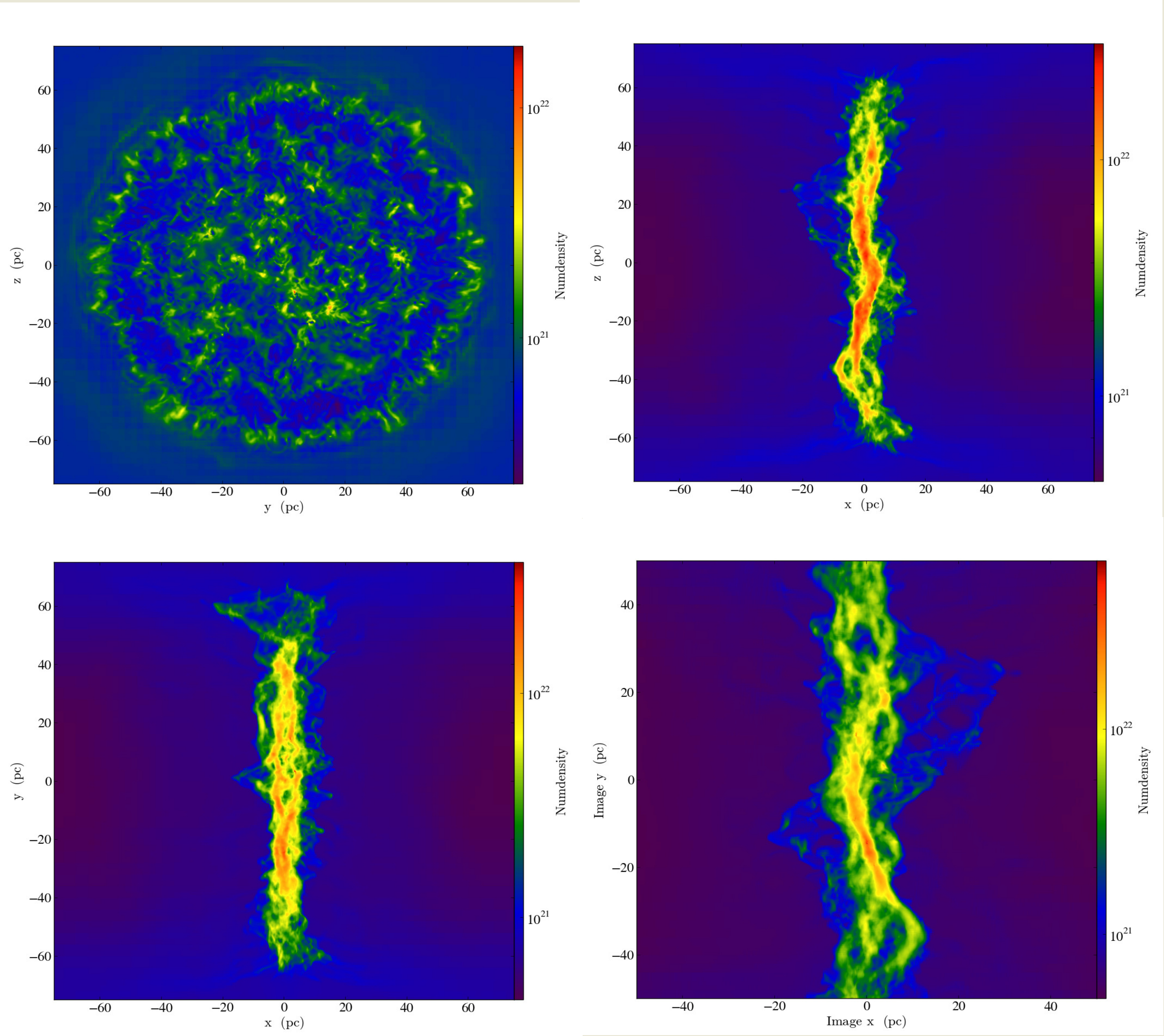

Fig. 1: Projection plots of the simulation box (256 pc^3) at $t \simeq 20$ Myr along each separate axis. A projection of a selected subvolume for a fixed angle $\theta = 84^\circ$ toward line-of-sight is plotted in bottom-right.

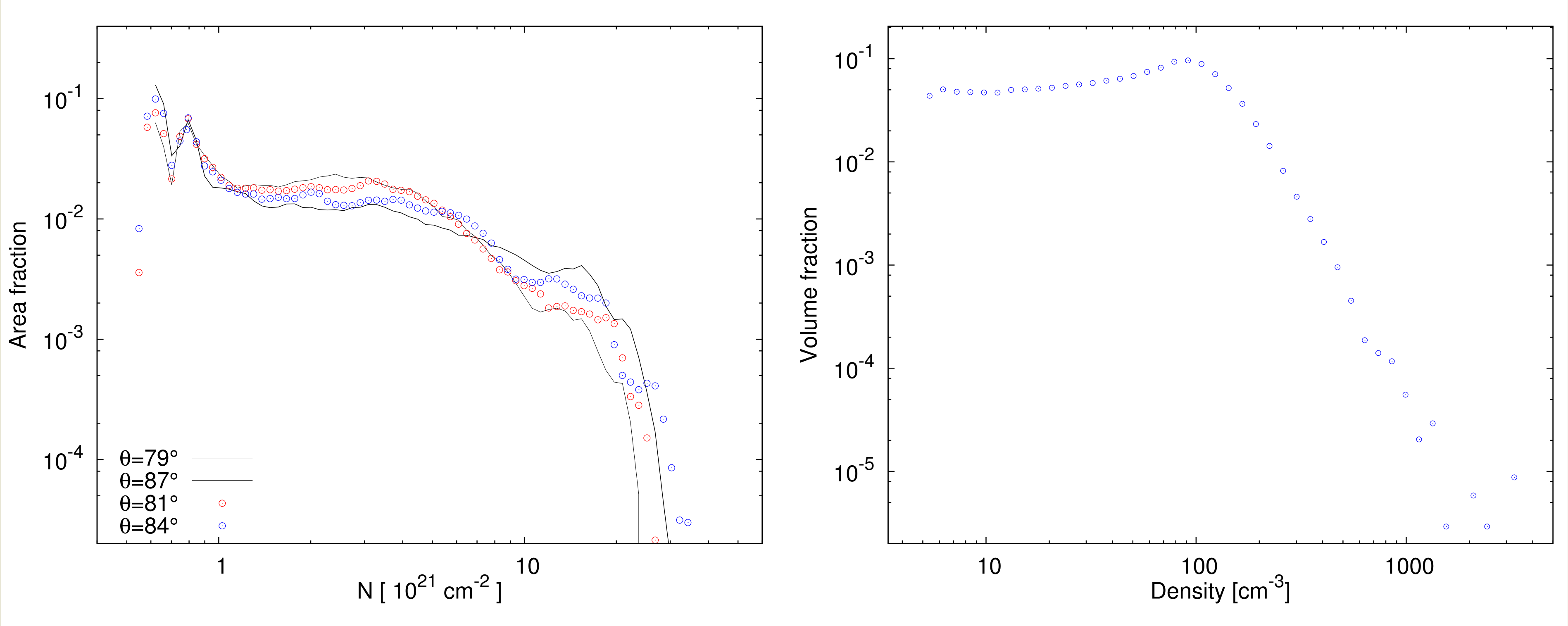

Fig. 2: *Left:* Column-density pdfs (N-pdfs) for delineated rectangular regions corresponding to several line-of-sight angles in the range $79 \leq \theta \leq 84$ degrees. The derived N-pdfs can be fitted through a combination of lognormal and power-law functions as demonstrated in other works (e.g. Vazquez-Semadeni 1994; Federrath & Klessen 2013). *Right:* A volume-density pdf of the dense gase phase (> 5 cm^{-3}) in the selected subvolume, shown in Fig. 1, bottom-right.

Source code for data projections: procedure

```
from yt.mods import *
from yt.utilities.physical_constants import cm_per_kpc, cm_per_mpc

parametricFile = load("run4_hdf5_plt_cnt_0390")

add_field("NumDensity", function=_NumDensity, units=r"\rm{cm}^{-3}")

angles = range(84, 85)
for angle in angles:
    project(parametricFile, angle, width = (100,'pc'), depth = (256, 'pc'))
```