Instituto de Física

INSTITUTO DE CIENCIAS FÍSICAS

IV TALLER DE MÉTODOS NUMÉRICOS Y ESTADÍSTICOS EN COSMOLOGÍA

30, 31 DE JULIO Y 1 DE AGOSTO

Benjamín Camacho Quevedo

Colaborando en el grupo SDSS-IV/eBOSS Galaxy Clustering con M. Vargas Magaña y Sebastien Fromenteau.

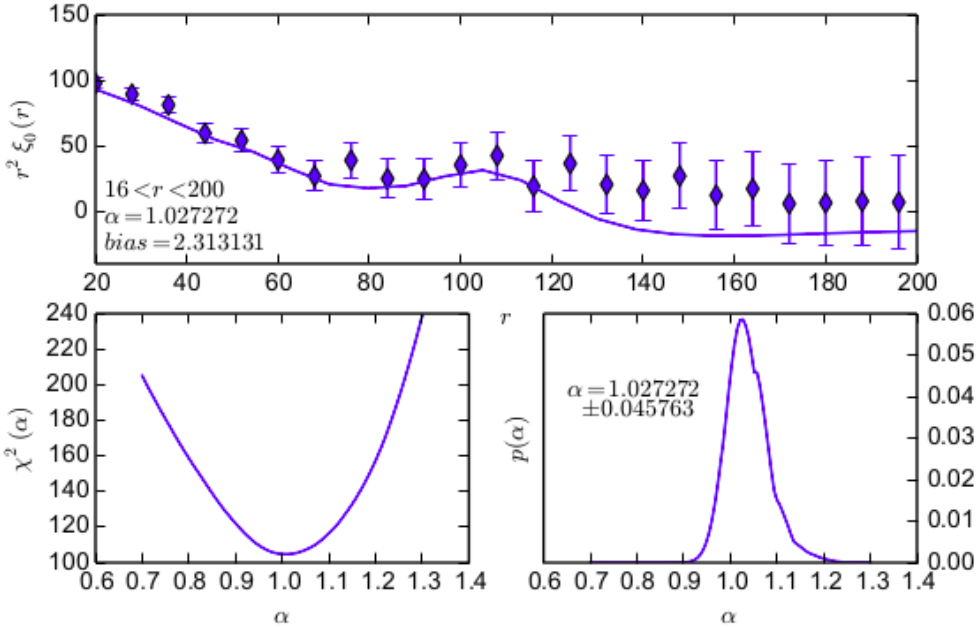
Motivación personal

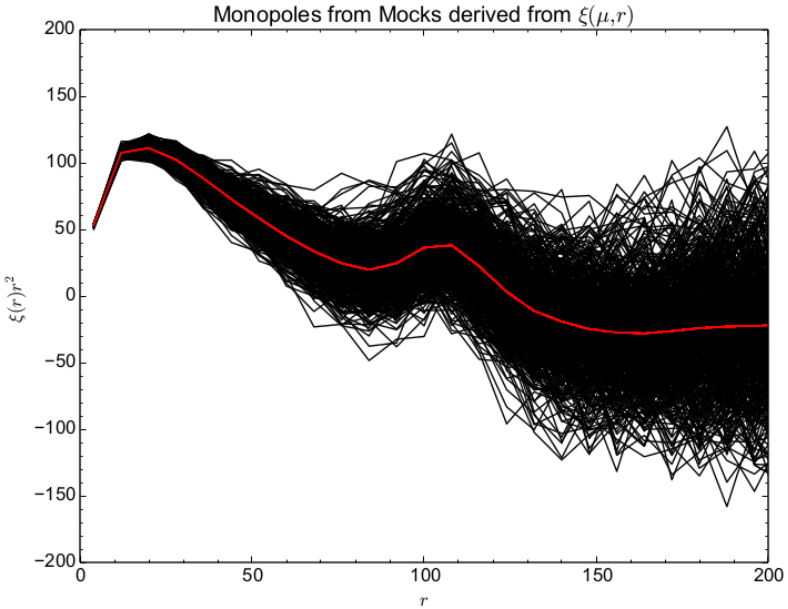
Era de los grandes sondeo cosmológicos.

Trabajo de tesis en la determinación de la escala de BAO con LRGs.

Preocupación por alcanzar la precisión requerida; determinación de la covarianza.

1000 catálogos necesarios para el análisis de LSS.





THE COMOVING LAGRANGIAN ACCELERATION (COLA) METHOD

Basic Idea

Standard case is to solve :

$$\partial_t^2 \vec{x} = -\vec{\nabla}\phi$$

COLA idea is to solve :

$$\vec{x} = \vec{x}_{LPT} + \vec{x}_{res}$$

$\mathcal{O}(\delta^3)$

$$\partial_t^2 \vec{x}_{res} = -\vec{\nabla}\phi - \partial_t^2 \vec{x}_{LPT}$$

Numerical discretization in a **PM** code

Analytic

MG-PICOLA =2LPT + N-body simulations

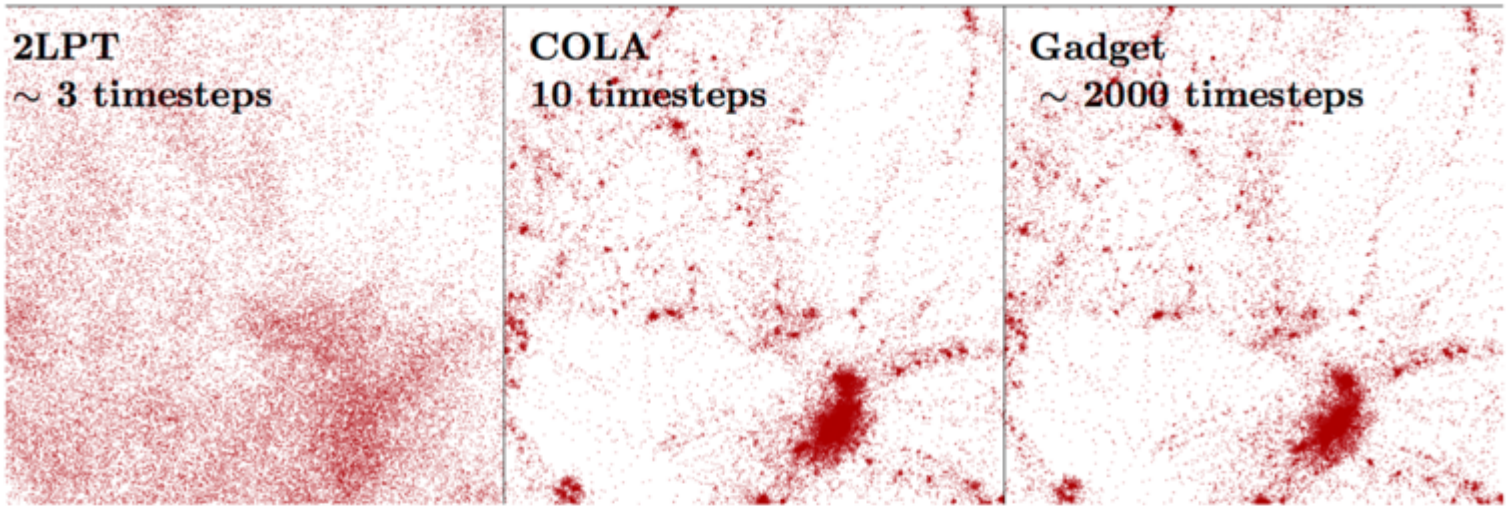


Figure 1: We show slices through three N-body simulations evolving the same initial conditions up to $z = 0$. The particles (each of mass $4.6 \times 10^9 M_\odot/h$) are shown as red points. Each slice is $20\,\mathrm{Mpc}/h$ on the side (the full simulation box is $100\,\mathrm{Mpc}/h$ on the side), and about $3\,\mathrm{Mpc}/h$ thick. The left panel shows the 2LPT approximation used for building mock catalogs using the PTHalos approach [7, 8]. Calculating the 2LPT particle positions requires an equivalent of roughly 3 timesteps performed by an N-body code. The middle panel shows the result obtained with our modified N-body code with as few as 10 timesteps. The rightmost panel shows the "true" result obtained from GADGET-2 [10] after ~ 2000 timesteps starting with 2LPT initial conditions at $z = 49$.

A LIGHTCONE-ENABLED PARALLEL IMPLEMENTATION OF COLA (L-PICOLA)

L-PICOLA

Sweet simulations with added caffeine

View on GitHub | Download .zip | Download .tar.gz

L-PICOLA is a distributed-memory, **parallel** code for creating **fast**, **accurate**, dark matter simulations.

- **How parallel?** We've used it for simulations with ~68,000,000,000 particles on over 1000 processors without a hitch.
- **How fast?** Due to the algorithms used and the coding, it is ~1000 times faster than an identical full N-Body simulation
- **How accurate?** We can reproduce the two- and three-point clustering of dark matter to within 2% on all scales used for current BAO and RSD measurements.

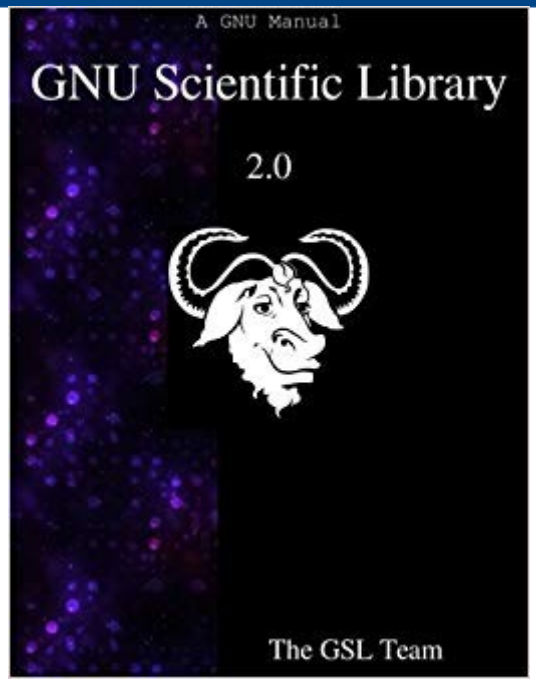
https://cullanhowlett.github.io/l-picola/

Howlett et al. 2015

Tassev et al. 2013,

Tassev et al. 2015

Prerequisites



https://www.open-mpi.org/

https://www.gnu.org/software/gsl/

http://www.fftw.org/

Go for it and unpack

- wget https://github.com/CullanHowlett/l-picola/tarball/master
- Git clone https://github.com/CullanHowlett/l-picola.git
- tar -zxvf master

Cullan Howlett Fixed bug in prepare_lightcone.f90 that stoppped compilation with gfo… ··· Latest commit dd385fe on May 13

Documentation	Added documentation detailing the new cosmological equation for L-PIC…	3 years ago
files	run_parameters and input_spectrum/transfer were inconsistent, so fine…	3 months ago
src	Fixed error in flag_replicates in lightcone.c, where the boundaries w…	5 months ago
utilities	Fixed bug in prepare_lightcone.f90 that stoppped compilation with gfo…	2 months ago
LICENSE	Create LICENSE	3 years ago
Makefile	Initial commit	3 years ago
README	Update README	3 years ago

Source

Branch: **master** | **l-picola** / **src** /

Create new file | **Find file** | **History**

Cullan Howlett Fixed error in flag_replicates in lightcone.c, where the boundaries w… ···

Latest commit 9407fff on Feb 19

..

2LPT.c	Bug fixes for runs in kpc instead of Mpc again. Previous correction o…	3 years ago
auxPM.c	Minor bug fix in output file name and changes to output units for par…	3 years ago
cosmo.c	Minor bug fix in output file name and changes to output units for par…	3 years ago
kernel.c	Bug fixes for runs in kpc instead of Mpc again. Previous correction o…	3 years ago
lightcone.c	Fixed error in flag_replicates in lightcone.c, where the boundaries w…	5 months ago
main.c	removed extraneous print statement in main.c	a year ago
power.c	Bug fixes for runs in kpc instead of Mpc again. Previous correction o…	3 years ago
proto.h	Minor bug fix in output file name and changes to output units for par…	3 years ago
read_param.c	Minor bug fix in output file name and changes to output units for par…	3 years ago
vars.c	Minor bug fix in output file name and changes to output units for par…	3 years ago
vars.h	Forgot to commit the vars.h and vars.c files for new GADGET header	a year ago

Files and Utilities

Branch: **master** | **l-picola** / **files** /

Create new file | Find file | History

Cullan Howlett run_parameters and input_spectrum/transfer were inconsistent, so fine… ··· Latest commit 036c063 on Apr 11

..		
input_kernel_equil.dat	Initial commit	3 years ago
input_kernel_local.dat	Initial commit	3 years ago
input_kernel_ortho.dat	Initial commit	3 years ago
input_spectrum.dat	run_parameters and input_spectrum/transfer were inconsistent, so fine…	3 months ago
input_transfer.dat	run_parameters and input_spectrum/transfer were inconsistent, so fine…	3 months ago
output_redshifts.dat	Initial commit	3 years ago
run_parameters.dat	run_parameters and input_spectrum/transfer were inconsistent, so fine…	3 months ago

Branch: **master** | **l-picola** / **utilities** /

Create new file | Find file | History

Cullan Howlett Fixed bug in prepare_lightcone.f90 that stoppped compilation with gfo… ··· Latest commit dd385fe on May 13

..		
L-PICOLA_mem.py	Initial commit	3 years ago
prepare_lightcone.f90	Fixed bug in prepare_lightcone.f90 that stoppped compilation with gfo…	2 months ago
prepare_snapshot.f90	Fixed bugs in gadget outputs in main.c and utilies/prepare_snapshot.f…	a year ago

Makefile Modifications

```
ifeq ($(MACHINE),ATOCATL)
  CC = mpicc
ifdef SINGLE_PRECISION
  FFTW_INCL = -I /opt/apps/libraries/gnu_4.4.6/fftw_3.3.3/include/
  FFTW_LIBS = -L /opt/apps/libraries/gnu_4.4.6/fftw_3.3.3/lib/  -lfftw3f_mpi -lfftw3f
else
  FFTW_INCL = -I /opt/apps/libraries/gnu_4.4.6/fftw_3.3.3/include/
  FFTW_LIBS = -L /opt/apps/libraries/gnu_4.4.6/fftw_3.3.3/lib/  -lfftw3_mpi -lfftw3
endif
  GSL_INCL  = -I /opt/apps/libraries/gnu_4.4.6/gsl_1.16/include/
  GSL_LIBS  = -L /opt/apps/libraries/gnu_4.4.6/gsl_1.16/lib/ -lgsl -lgslcblas
  MPI_INCL  = -I $/opt/apps/clustertools/gnu_4.8.5/ompi_2.1.1/include
  MPI_LIBS  = -L $/opt/apps/clustertools/gnu_4.8.5/ompi_2.1.1/lib -lmpi
endif
```

How to look for them?

- module Av
- module load library
- echo $LIBRARY_HOME

And then...

- Make
- mpirun -np 24 L-PICOLA /files/run_parameters.dat

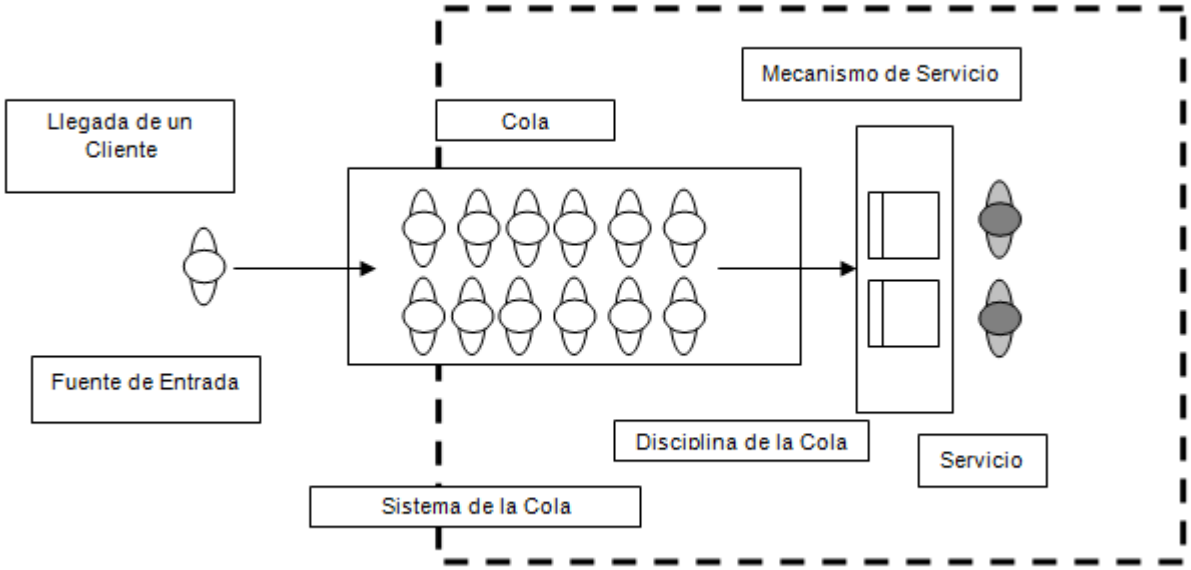
Input

```
% ============================== %
% This is the run parameters file %
% ============================== %

% Simulation outputs
% ==================
OutputDir                   files/                          % Directory for output.
FileBase                    example_filename                % Base-filename of output files (appropriate additions are appended on at runtime)
OutputRedshiftFile          files/output_redshifts.dat      % The file containing the redshifts that we want snapshots for
NumFilesWrittenInParallel   4                               % limits the number of files that are written in parallel when outputting.

% Simulation Specifications
% =========================
UseCOLA          1           % Whether or not to use the COLA method (1=true, 0=false).
Buffer           1.3         % The amount of extra memory to reserve for particles moving between tasks during runtime.
Nmesh            256         % This is the size of the FFT grid used to compute the displacement field and gravitational forces.
Nsample          256         % This sets the total number of particles in the simulation, such that Ntot = Nsample^3.
Box              512.0      % The Periodic box size of simulation.
Init_Redshift    9.0         % The redshift to begin timestepping from (redshift = 9 works well for COLA)
Seed             5001        % Seed for IC-generator
SphereMode       0           % If "1" only modes with |k| < k_Nyquist are used to generate initial conditions (i.e. a sphere in k-space),
                             % otherwise modes with |k_x|,|k_y|,|k_z| < k_Nyquist are used (i.e. a cube in k-space).

WhichSpectrum    1           % "0" - Use transfer function, not power spectrum
                             % "1" - Use a tabulated power spectrum in the file 'FileWithInputSpectrum'
                             % otherwise, Eisenstein and Hu (1998) parametrization is used
                             % Non-Gaussian case requires "0" and that we use the transfer function

WhichTransfer    0           % "0" - Use power spectrum, not transfer function
                             % "1" - Use a tabulated transfer function in the file 'FileWithInputTransfer'
                             % otherwise, Eisenstein and Hu (1998) parameterization used
                             % For Non-Gaussian models this is required (rather than the power spectrum)

FileWithInputSpectrum  files/input_spectrum.dat    % filename of tabulated input spectrum (if used)
                                                   % expecting k and Pk

FileWithInputTransfer  files/input_transfer.dat    % filename of tabulated transfer function (if used)
                                                   % expecting k and T (unnormalized)
```

Input

```
% Cosmological Parameters
% ======================
Omega            0.31        % Total matter density (CDM + Baryons at z=0).
OmegaBaryon      0.048        % Baryon density (at z=0).
OmegaLambda      0.69        % Dark Energy density (at z=0)
HubbleParam      0.69         % Hubble parameter, 'little h' (only used for power spectrum parameterization).
Sigma8           0.83        % Power spectrum normalization (power spectrum may already be normalized correctly).
PrimordialIndex  0.96        % Used to tilt the power spectrum for non-tabulated power spectra (if != 1.0 and nongaussian, generic flag required)

% Timestepping Options
% ====================
StepDist         0           % The timestep spacing (0 for linear in a, 1 for logarithmic in a)
DeltaA           0           % The type of timestepping: "0" - Use modified COLA timestepping for Kick and Drift. Please choose a value for nLPT.
                             % The type of timestepping: "1" - Use modified COLA timestepping for Kick and standard Quinn timestepping for Drift.
Please choose a value for nLPT.
                             % The type of timestepping: "2" - Use standard Quinn timestepping for Kick and Drift
                             % The type of timestepping: "3" - Use non-integral timestepping for Kick and Drift
nLPT             -2.5        % The value of nLPT to use for modified COLA timestepping


% Units
% =====
UnitLength_in_cm                3.085678e24       % defines length unit of output (in cm/h)
UnitMass_in_g                   1.989e43          % defines mass unit of output (in g/h)
UnitVelocity_in_cm_per_s        1e5               % defines velocity unit of output (in cm/sec)
InputSpectrum_UnitLength_in_cm  3.085678e24       % defines length unit of tabulated input spectrum in cm/h.
                                                  % Note: This can be chosen different from UnitLength_in_cm
```

Sistema de colas



ATOCATL es un sistema de supercómputo híbrido CPUs/GPUs para usarse con (i) códigos que requieren únicamente de MPI/OpenMP , o bien, (ii) para códigos que también pueden obtener ventaja de la arquitectura de las Unidades de Procesamiento Gráfico (GPUs) y también para códigos seriales. Es un esfuerzo colectivo/institucional por proveer a la comunidad del IA-UNAM de una herramienta de gran poder de supercómputo para realizar tanto simulaciones como procesamiento de bases de datos.

Sistema de colas (PBS)

Se han incorporado colas que reflejan la versatilidad de las arquitecturas presentes en Atocatl. Las colas se dividen de la siguiente manera:

Las siguientes colas tienen disponibilidad de hasta 1 semana de wall-clock.

sq1i - cola para corridas en serie en procesadores Intel (compute-0-5, compute-0-9 y compute-0-10)

pq1a - cola para corridas paralelas en procesadores AMD (compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-4 y compute-0-6).

pq2a - cola para corridas paralelas en procesadores (compute-0-7)

pq2a_d - cola para corridas paralelas en procesadores de los nodos de Dany (compute-0-7 y compute-0-8)

gpuq1i - cola para corridas en GPUs en nodos con procesadores Intel (compute-0-9 y compute-0-10)

gfq1i - cola para corridas con procesadores Intel dentro del proyecto de Vladimir (compute-1-2)

desiq - cola para corridas utilizando cualquiera o todos los procesadores de los nodos del proyecto DESI (compute-1-0 y compute-1-1, compute-1-3, compute-1-4 y compute-1-5)

desiq1i - cola para corridas utilizando cualquiera o todos los procesadores Intel de los nodos del proyecto DESI (compute-1-3, compute-1-4 y compute-1-5)

desiq1a - cola para corridas utilizando cualquiera o todos los procesadores AMD de los nodos del proyecto DESI (compute-1-0 y compute-1-1)

La cola "**default**" no está restringida en cuanto al tipo de arquitectura pero si al tiempo de procesamiento que se limita a 25 minutos únicamente.

Example of a job

```
#! /bin/bash
#PBS -V
#PBS -N L-PICOLA1
#PBS -q sq1i
#PBS -S /bin/sh
#PBS -o test1.out
#PBS -e test1.err
#PBS -l nodes=1:ppn=12

module load openmpi/2.1.1_gnu
module load gsl/1.16_gnu
module load fftw/3.3.3_ompi_gnu

mpirun -np 12 /home/bcamacho/L-PICOLA/CullanHowlett-l-picola-dd385fe/L-PICOLA /home/bcamacho/L-PICOLA/CullanHowlett-l-picola-dd385fe/files/run_parame\
ters.dat
```

Comandos útiles

El envió de un proceso al sistema de colas se puede realizar a través de un script en el cual se deben especificar: la cola, el número de nodos, el número de cores por nodo, etc.

Algunos comandos útiles para checar los procesos en el sistema de colas:

%> qstat -u Nombre_del_Usuario - Proporciona información sobre el o los procesos del usuario.

%> qstat -q - Proporciona información sobre los procesos en general en el sistema de colas.

%> showq - Proporciona información sobre la cantidad total de cores en uso.

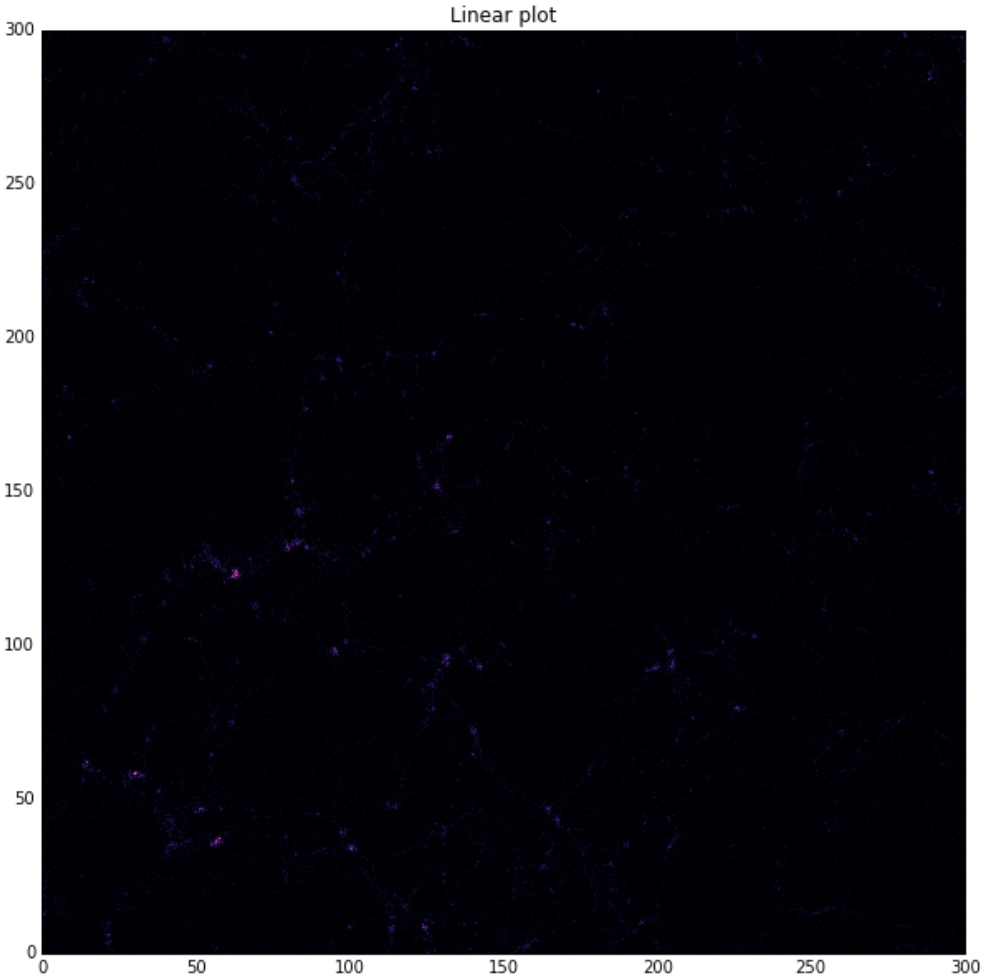
%> qsub -V mi_script - Comando para el envio del proceso al sistema de colas. La opción "-V" es necesaria para transmitir las variables de ambiente a todos los nodos que usará el usuario para su corrida, a menos que la haya incluido dentro del script "mi_script". El sistema de colas le asignará un ID al proceso (Job ID o JOBNAME) que puede ser de utilidad para ir monitoreando el desarrollo del proceso. Para enviar el proceso es necesario que tenga cargado el o los módulos utilizados para compilar su código.

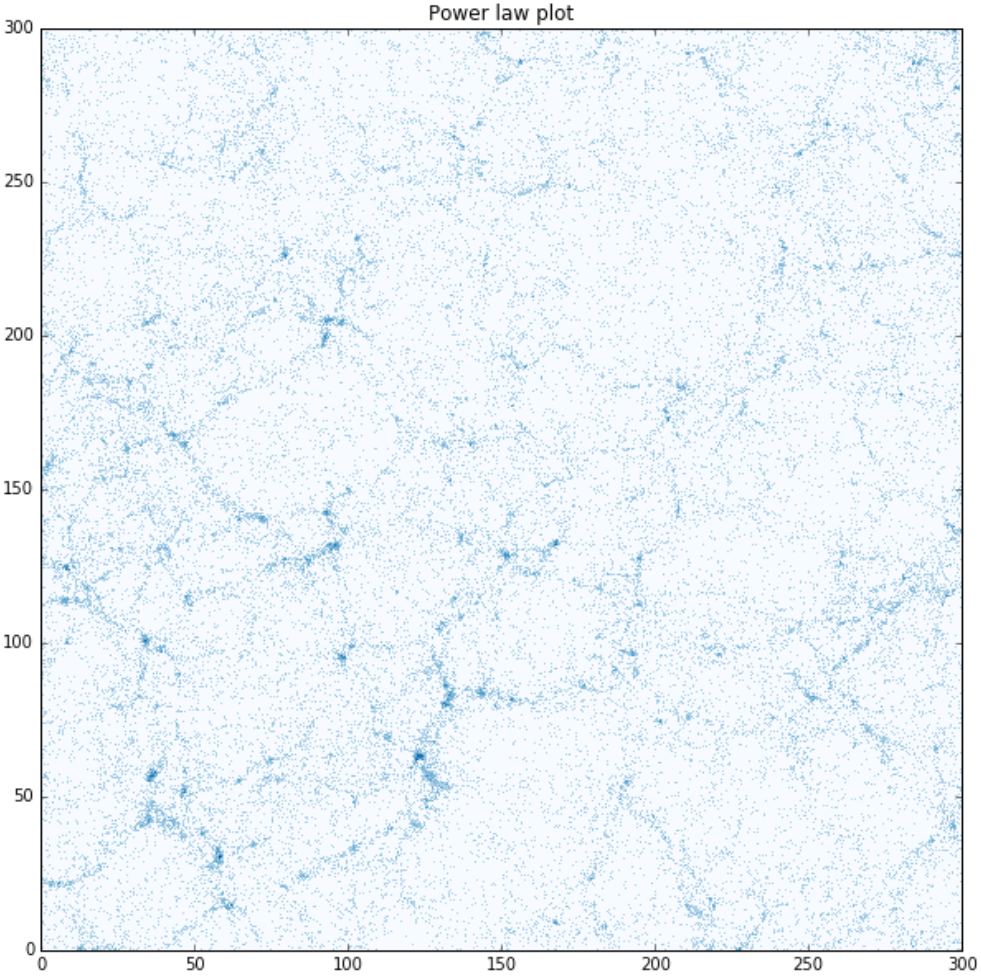
%> canceljob ID_del_proceso - Interrumpe y libera los recursos de la cola utilizados por el proceso.

Output


GADGET - 2
A code for cosmological simulations of structure formation

Visualization (Pynbody)





Script preparado para el IV TALLER DE MÉTODOS NUMÉRICOS Y ESTADÍSTICOS EN COSMOLOGÍA.

In [1]:
```
##### Notebook to read 1 file from gadget and do a plot of the density field

import matplotlib.pyplot as plt
%matplotlib inline
import pynbody


##### Path to the file
path_data = '/home/bcamacho/Desktop/eBOSS/LSSanalysis-master/nbodikit/'


##### Example to rad a file named 'mysimu_z0p500.0' which correspond in this case of:
##### "mysimu" is the name you give in the parameterfile your run for the simu
##### "_z0p500" correspond to a snapshot at z=0.5
##### ".0" is the first chunk of the simu. The simu has 1 chunk for each CPU used

#file_name = path_data +'mysimu_z0p500.0'
file_name = path_data+'2example_filename_z0p000.0'
#### Read the gadget format from the COLA output
s = pynbody.load(file_name)

#### We create the x, y and z array containning the positions for all the particles in this chunk

x = (s['pos']).T[0]
y = (s['pos']).T[1]
z = (s['pos']).T[2]
```

In [2]:
```
import numpy as np

##### We want to know the minima and maxima in x, y and z dimensions for this chunk

print "Xmin =",  np.min(x), "Xmax=", np.max(x)
print "Ymin =",  np.min(y), "Ymax=", np.max(y)
print "Zmin =",  np.min(z), "Zmax=", np.max(z)
```

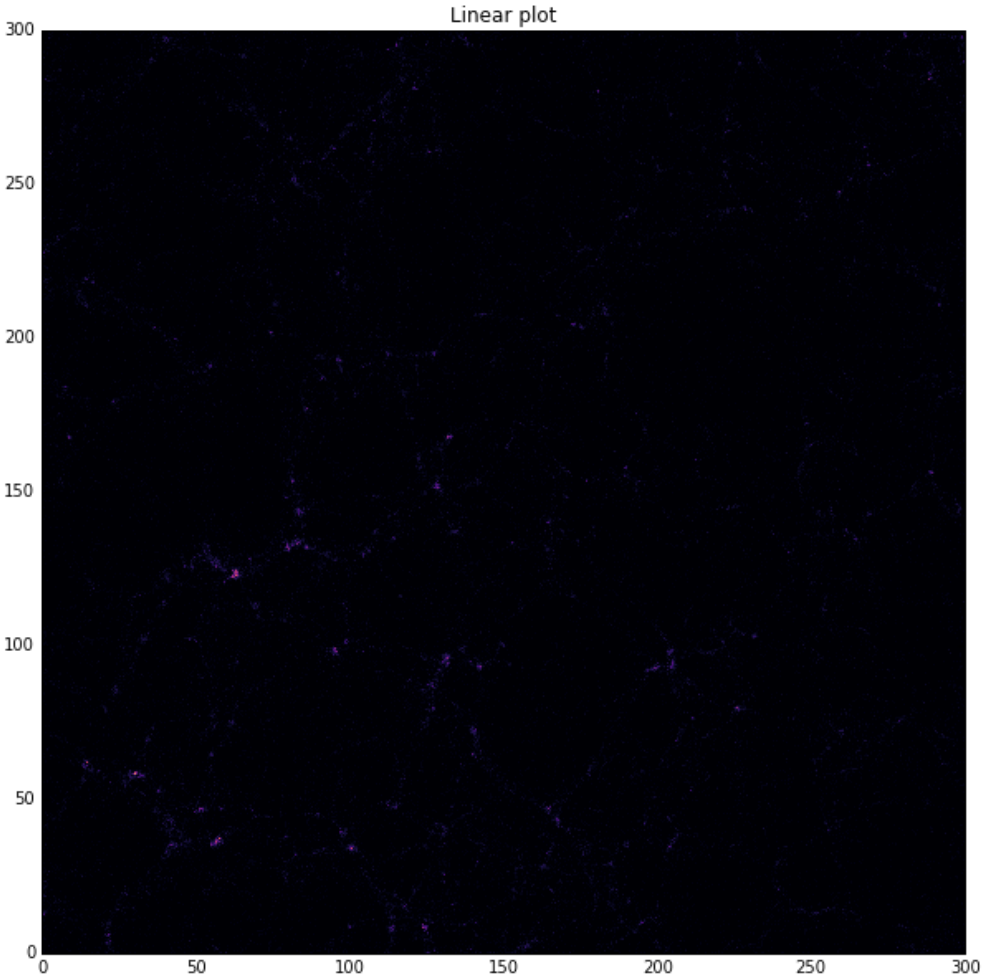
```
Xmin = 0.0 Xmax= 127.99995
Ymin = 0.0 Ymax= 511.99988
Zmin = 0.0 Zmax= 511.9998
```

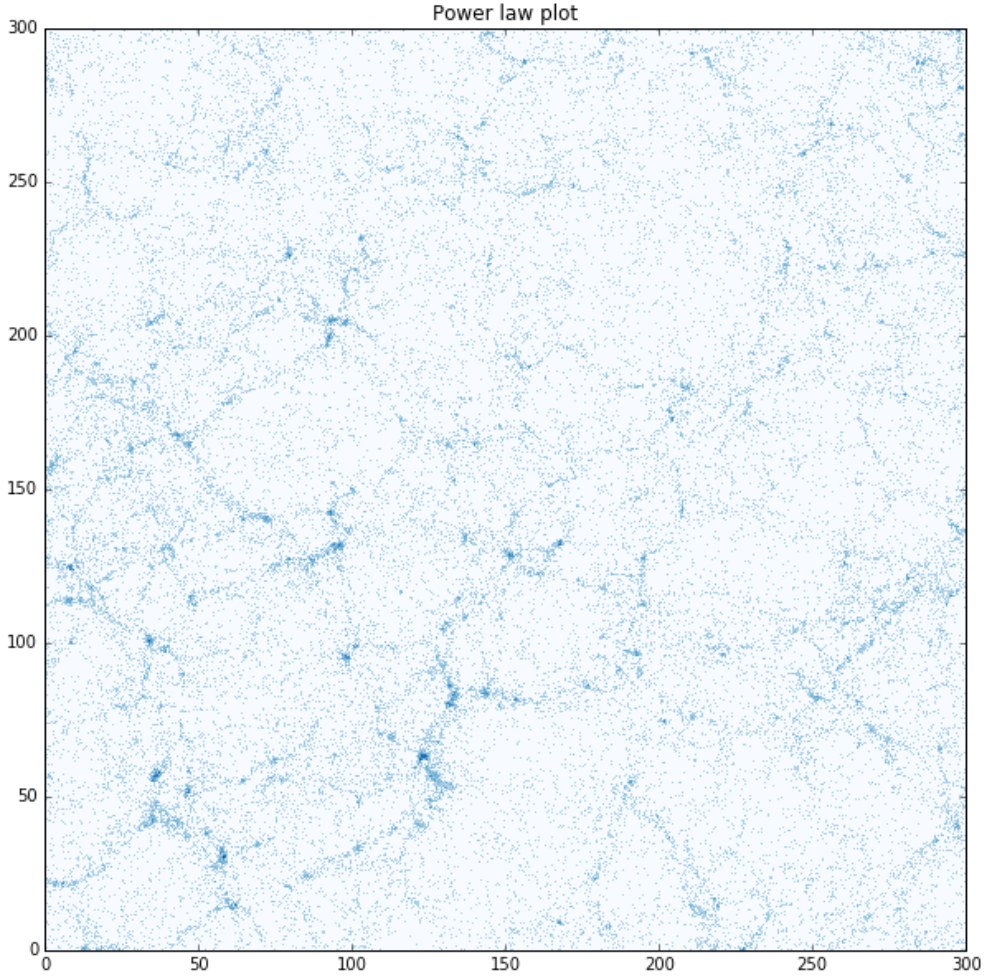
In [6]:
```
##### Now we have the information we can choose a slice inside this chunk we want to show
##### Here I will choose a slice of 30 Mpc width in X and 300 by 300 Mpc in Y and Z
##### You have to adapt these cuts for your simulation.


##### Cuts to represent a slice with 30Mpc width in X
xmin = 10
xmax = 40

ymin = 100
ymax = 400

zmin = 100
zmax = 400


#### We select the particles inside the bounds we defined
tmp = np.where( (x < xmax) & (x > xmin) & (y < ymax) & (y > ymin) & (z < zmax) & (z > zmin) )

tmp = tmp[0]

x = x[tmp]
y = y[tmp]
z = z[tmp]
```

In [7]:
```
#### We have to fill a 2D array in which we will stack the particles in the X dimension.
#### So we will have a 2D array of 300Mpc by 300Mpc (Y,Z)

L_box = 300
pix_size = 0.2  ##### size of the pixels in Mpc. You can play with this parameter
N_pix = int(L_box/pix_size) +1   ##### calculate the number of pixels in each dimension

y_arr = np.linspace(0,L_box, N_pix)  #### create the array with the value of the pixel position in Y
z_arr = np.linspace(0,L_box, N_pix)  #### create the array with the value of the pixel position in Z

#### density is the 2D array we will fill
density = np.zeros( (N_pix, N_pix) )

##### We find the cell of density array hosting each of the particles and we add 1 in it
for i in range(len(y)):
    ind_x = int(np.floor( (y[i]-ymin) /pix_size))
    ind_y = int(np.floor( (z[i]-zmin)/pix_size))
    density[ind_x, ind_y] += 1.
```

In [8]:
```
#### And we can plot the results. Of course we do not plot in linear scale
#### because we do not see anything
plt.figure(figsize=(10,10))
plt.pcolormesh(y_arr, z_arr, (density), cmap='magma' )
plt.title("Linear plot")
plt.show()
```

```
plt.title("Power law plot")
plt.show()
```


Linear plot
300
250
200
150
100
50
0
0
50
100
150
200
250
300


Power law plot
300
250
200
150
100
50
0
0
50
100
150
200
250
300