

FOFReaderLib

Abstract

FOFReadLib is a C++ Library to read the FOF files results of the cosmology simulations done by the DEUS Consortium (Dark Energy Universe Simulation)

<http://www.deus-consortium.org/>

Those files are Fortran generated binaries, some in Little Endian, other in Big Endian.

Thanks to this library, the data can be manipulated in an object programming manner without having to know the exact source format.

Installation

A simple Makefile is provided.

Prerequisites

GNU g++ version 4.8.x

It may be working with other versions but has not been tested.

Configuration

The following options are used for g++ in the Makefile

```
-c -Wall -O3 -g -fopenmp -std=c++0x
```

Features

The best way to understand this library is to read sample programs source code.

FOFFiles

FOFFiles are low level reading objects.

FOFMultiCube Can read Cube and MultiCube files, ie the particles.

Each cube is a subpart of a snapshot and have area properties (min/max of x,y,z)

The object can take either a single file or a full directory as input.

If single file is provided, default is to read Position and Velocity of particles. If directory is provided, default is not to read particles.

Particles can be read/released on demand to manage memory. Options for reading are DONT_READ_PARTICLES, READ_POS, READ_VEL, READ_IDS. READ_ALL is an alias for READ_POS | READ_VEL | READ_IDS.

See FOFReader.cpp as example, format 0.

FOFStrct Can read Strct files, ie particles from the halos.

Particles can be read/released on demand to manage memory in the same way than cube files.

Strct format is not easy to browse, to select halos easily before reading particles, see higher level object DEUSHalos.

See FOFReader.cpp as example, format 1.

FOFMasst Can read Masst files, ie center of mass and npart of halos.

Masst is easy to browse but doesn't contain particle data, see DEUSHalos to mix masst and strct.

See FOFReader.cpp as example, format 2.

FOFCubeGrav Can read Cube grav files, ie the grid with the different refinement levels.

The full structure with number of levels is read. To read level content, a readLevels() call is needed. By default it only reads coarse level.

This simple object can only read files. For directories see DEUSGrav.

See FOFReader.cpp as example, format 4.

DeusSimulation

DeusSimulation contains higher level tools based on FOFFiles.

DEUSArea Tools to manage areas: can find cubes and particles belonging to an area. Can manage periodicity.

See getAreaFiles.cpp as example.

DEUSGrav Tools to manage Grav files: can read a full directory and browse among the FOFGrav.

DEUSHalos Tools to manage halos: use Masst and Strct files together. Use Masst for fast index, and read Strct on demand to get particles.

See FOFReader.cpp as example, format 3.

DEUSSimulationSingleton High level object: can read at the same time cubes and halos to manage a full simulation.

Sample Programs

FOFReader

Sample program to read the different available formats and display some infos.
Can read either files or full directories

```
Usage : FOFReader <file> <format>
      Formats:
      0: FOF Cube/Multicube
      1: FOF Struct
      2: FOF Masst
      3: DEUS Halo directory (= masst + strct)
      4: FOF Grav cube
```

exportMasst

Simple utility to transcode Masst Fof files into simple Ascii text

```
Usage : exportMasst <directory> <output.txt>
```

getBiggestHalo

Simple utility to get biggest halos from Masst files

```
Usage: getBiggestHalo full_path [nbTops]
Example: getBiggestHalo /mySimulation/post/fof/output00025/ 5
```

generateRandomCube

Generate random cube files for test purpose

Usage : `generateRandomCube <filename> <lpart> <generator>`
will generate a cube with `lpart*lpart*lpart` particles using generator:
0: constant, all particles in (0.5,0.5,0.5) with velocity(0,0,0)
1: random, all particles with random position and velocity between 0.0 and 1.0
2: homogeneous, all particles in a constant grid with velocity(0,0,0)

getAreaFiles

List the cube files needed for a subregion

Usage : `getAreaFiles <directory> <center_x> <center_y> <center_z> <size> [periodicity]`
center_x / center_y / center_z : center to get in float
center of snapshots is 0.5 0.5 0.5
center of cone (ie Z=0) is 0.0 0.0 0.0
size : size to get in float (max 1.0)
periodicity : 1.0 for snapshots 0.0 (none, default) for cone

generateImage

Extract a slice from Grav Cube files and generate a picture file in “gray format”
readable by imagemagick

Usage : `generateImage <directory> <output.gray>`

Licence

This library has been created by Jean Pasdeloup under CeCILL-B Licence

It includes some code from Vincent Reverdy (Fortranfile.h) also under CeCILL-B
Licence