Creation of initial conditions

Galstep

Creates initial conditions for galaxies.

First version was made by Raffael Ruggiero, but I changed some parameters and tweaked the code in order to write for Gadget-4 HDF5 and work in python3.

My version can be found here and quickly downloaded using the terminal with

```
wget https://github.com/elvismello/galstep/archive/refs/heads/master.tar.gz
```

Clustep

Creates initial conditions for galaxy clusters.

First version was made by Raffael Ruggiero, but I changed some parameters and tweaked the code in order to write for Gadget-4 HDF5 and work in python3.

My version can be found here and quickly downloaded using the terminal with

```
wget https://github.com/elvismello/clustep/archive/refs/heads/master.tar.gz
```

snapshotJoiner

It joins snapshots.

Code can be found here.

```
wget
https://github.com/elvismello/snapshotJoiner/archive/refs/heads/master.tar.
gz
```

Programs for manually reading and writing initial conditions

UNSIO

A suitable substitute for `pygadgetreader` and (possibly) `pynbody` for reading Gadget-2 snapshots. It can also be used to write snapshots, which can be quite useful.

It was made by the same author of Glnemo2, Jean.

```
pip install python-unsio
```

There are versions for Fortran and C/C++.

Scripts

Rubens has written some basic scripts that use UNSIO specifically with Gadget-2, which can be found here.

To automatically get a tarball:

```
wget
https://github.com/regmachado/SnapGadget/archive/refs/heads/main.tar.gz
```

Formats

Can be used to read

- Gadget 1;
- Gadget-2;
- Gadget-3 (hdf5);
- Nemo;
- Ramses;
- Misc (List of files, sqlite3 database);

and write

- Gadget-2;
- Gadget-3 (hdf5);
- Nemo.

User guides

For reading files:

- https://projets.lam.fr/projects/unsio/wiki/PythonReadDataNew

For writing files:

- https://projets.lam.fr/projects/unsio/wiki/PythonWriteDataNew

Quick reference

Components present in snapshots

Component tag	Description

Component tag	Description
gas	Gas particles
halo	Dark matter particles
dm	Dark matter particles
disk	Old stars particles
stars	Stars particles
bulge	Bulge particles
bndry	bndry particles

tag string	descripton	numpy data_array passed as parameter
pos	particles positions	numpy 1D array of particles position (size=n*3)
vel	particles velocities	numpy 1D array of particles velocitie (size=n*3)
mass	particles masses	numpy 1D array of particles velocitie (size=n)
acc	particles accelerations	numpy 1D array of particles acceleration (size=n*3)
pot	particles potential	numpy 1D array of particles potential (size=n)
rho	particles densities	numpy 1D array of particles density (size=n)
hsml	particles hsml	numpy 1D array of particles hydro smooth length (size=n)
temp	particles temperatures	numpy 1D array of particles temperature (size=n)
age	particles ages	numpy 1D array of particles age (size=n)
metal	particles metallicity	numpy 1D array of particles metallicity (size=n)
u	particles internal energy	numpy 1D array of particles internal energy (size=n)
aux	particles auxilliary (NEMO)	numpy 1D auxiliary array (size=n)
keys	particles keys (NEMO)	numpy 1D keys array (size=n)
id	particles indexes (NEMO)	numpy 1D keys array (size=n)

Pynbody

A recent development that is supposed to be "The One Program to Analyze Them All". It doesn't always work as intended for non-cosmological Gadget simulations (because of units) and wrongly recognizes Gadget-4 HDF5 as Arepo snapshots. Can write basic snapshots in some formats.

It is still quite simple to be used and can produce quick results and great visualizations, being useful in a pinch.

It can also run with multiple threads and has many sections written in C/C++, being surprisingly fast even though it uses mainly Python.

```
pip install pynbody
```

User guides

Tutorials:

- https://pynbody.github.io/pynbody/tutorials/tutorials.html

Pynbody framework specifics:

- https://pynbody.github.io/pynbody/reference/essentials.html

H5PY

Python API that interfaces with HDF5 libraries. Can be used for reading and writing in HDF5.

Most of the time, it is as dificult to use as pynbody or unsio.

```
pip install h5py
```

Warnings

- Sometimes Gadget-4 snapshots written in the binary format will have strange fields that not every program will be able to read.
- Even though Gadget-4 understands the binary format from Gadget-2, it will not accept it as initial conditions, unless a block of metallicity is inserted into the format for gas particles (I think). Python 3 versions of galstep and clustep will usually work.
    - UNSIO has a way to use this block, reading and writing to it, but didn't always work the last time I checked.
- HDF5 snapshots from Gadget-2/3 and Gadget-4 are not exactly made in the same way. There are different HDF5 groups for the header and other parameters, *BUT* essential information, such as particle positions and velocities, are stored in the same way.

Auxiliary programs and commands

Vitables

Usefull for messing with HDF5 files. I recomend always opening snapshots in "read-only".

```
sudo apt install vitables
```

ssh

It is used for connecting to an external server by using the terminal. Comes by default with most linux distros and mac systems.

Exemplo:

```
ssh <user>@<host>
```

If the username from the connecting computer is the same as the account being connected in the server, just

```
ssh <host>
```

is enough.

ssh keys

Sometimes, when downloading/uploding numerous files from a ssh connection, it gets cumbersome to always type the password. For this, you can generate a pair of keys that, when installed, will eliminate the need to always enter the password when connecting to the ssh server.

First the keys are created (a usual path is `~/.ssh`)

```
ssh-keygen -f <path>/<key_pair_name>
```

Then the keys are copied and installed onto the server

```
ssh-copy-id -i <path>/<key_pair_name> <user>@<host>
```

Now, depending on server settings and key settings, there shouldn't be the need to repeatedly type the password.

Author

Elvis de Mello

Github: https://github.com/elvismello

e-mail: elvis.mello@outlook.com / elvis.mello@ufpr.br