User Guide

Shihong Liao

July 11, 2018

Contents

1 Introduction

To generate an initial condition for a cosmological N-body simulation, we need to set up a uniform distribution of N particles before applying the Lagrangian perturbation theory to perturb their positions and velocities according to a given matter power spectrum. Such a uniform particle distribution is called a *pre-initial condition* or a *particle load*, and it can be generated by several known methods such as the grid (e.g., Efstathiou et al., 1985), glass (White, 1996) and Quaquaversal (Hansen et al., 2007) methods.

This code adopts an alternative method, the capacity constrained Voronoi tessellation (CCVT, Balzer et al., 2009), to generate pre-initial conditions. It is written in C, and uses OpenMP to accelerate its speed. It uses Qhull (Barber et al., 1996) to perform Voronoi and Delaunay tessellations. It also provides an option to accelerate the computations by using the capacity constrained Delaunay triangulation (CCDT, Xu et al., 2011) to initialize the particle distribution. The details and tests of this code are described in Liao (2018).

This code is free software. You can redistribute it and/or modify it under the terms of the GNU General Public License.

2 Code downloads

You can directly download this code from https://github.com/liaoshong/ccvt-preic, or you can use git to clone it,

```
$ git clone https://github.com/liaoshong/ccvt-preic.git
```

3 Environment requirements

To compile the program, you need an ANSI C compiler with OpenMP supported (e.g. GCC later than version 4.2).

4 How to use?

1. **Edit the configuration options in `configure.h`.** Currently, two options are provided in `configure.h`:

    - `DIMENSION`: The dimension of the CCVT configuration you want to generate. Currently, only 2D and 3D are supported. To generate pre-initial conditions for cosmological N-body simulations, `DIMENSION` should be set to 3.

    - `CCDT_INIT`: Switch on/off the CCDT initialization option. 1: switch on; otherwise: switch off. Recommend to switch on it to accelerate the computation.

2. **Compile the code.** Type

    ```
    $ make
    ```

    in the command line, and normally it should generate an executable named `ccvt-preic`.

3. **Edit the parameter file.** There is an example parameter file in `results/` with a file name of `run.param`. Edit the listed parameters whose meanings are summarized as follows.

    - `NUM_PART_EACH_DIM`: The number of particles on each dimension. The total number of particles is thus equal to $\texttt{NUM\_PART\_EACH\_DIM}^d$, where $d = $ `DIMENSION`. The given `NUM_PART_EACH_DIM` should be an integer larger than `1`. It has been shown in Liao (2018) that it is valid to use a tiling CCVT in cosmological simulations. Therefore, to make a large CCVT, you can first generate a smaller periodic CCVT, and then use tilings when set up initial conditions for cosmological simulations. See Section 6 for an example about how to set the tiling factor in initial condition generators.
    - `NUM_CAPACITY_EACH_DIM`: The number of capacity of each particle on each dimension. Thus, the capacity of each particle is equal to $\texttt{NUM\_CAPACITY\_EACH\_DIM}^d$. The input `NUM_CAPACITY_EACH_DIM` should be an integer larger than `1`. Usually, a value between 10 and 20 should be adopted for `NUM_CAPACITY_EACH_DIM` to obtain a good CCVT pre-initial condition; see Liao (2018) for detailed convergence tests.

- BOX_SIZE: The box size of the generated particles. Usually, it is set to 1.0 or other positive float number since the final configuration can be scaled to any box size.
- RANDOM_SEED: The integer seed for generating pseudo random numbers.
- NUM_THREADS: The number of threads for OpenMP parallelizations.
- OUTPUT_PATH: The path to save outputs.

4. **Run the code with the below command.**

```
$ ./ccvt-preic results/run.param
```

5 Outputs

The code outputs two files:

- ccvt_particle_[NUM_PART_EACH_DIM]_capacity_[NUM_CAPACITY_EACH_DIM].txt: a text file with particles' positions.

- gadget_particle_[NUM_PART_EACH_DIM]_capacity_[NUM_CAPACITY_EACH_DIM]: a file with the GADGET format (i.e., SnapFormat = 1 in GADGET's conventions). This file can be directly used by the N-GenIC or 2LPTic codes (see Section 6 for details). Note that this file is output only when DIMENSION = 3.

If you prefer to generate files with other formats, you can edit output.c.

6 Use in N-GenIC and 2LPTic

To use CCVT pre-initial conditions in N-GenIC (or 2LPTic), you need to switch on the CORRECT_CIC option in the Makefile, and specify two parameters in the N-GenIC (or 2LPTic) parameter file:

- GlassFile: the path of your CCVT pre-initial condition (i.e., path to gadget_particle_[NUM_PART_EACH_DIM]_capacity_[NUM_CAPACITY_EACH_DIM]).

- GlassTileFac: The tiling factor of the input CCVT configuration. It equals to Nsample/NUM_PART_EACH_DIM. For example, if you want to generate an initial condition with 64^3 particles (Nsample = 64) using N-GenIC (or 2LPTic), and you have a CCVT pre-initial condition with 16^3 particles (NUM_PART_EACH_DIM = 16), then you should set GlassTileFac = 64/16 = 4.

7 Data downloads

Some CCVT pre-initial conditions, which cost more computational time to compute, are provided at https://github.com/liaoshong/ccvt-preic-data. You can download them in the webpage, or use git to clone them:

```
$ git clone https://github.com/liaoshong/ccvt-preic-data.git
```

This data repository may add more data in the future.

8 Contacts

Questions and bug reports can be sent to Shihong Liao (liaoshong@gmail.com). Feedbacks and comments are welcome.

9 Acknowledgements

The author thanks Jianxiong Chen for his feedbacks and suggestions on this code, and Michael Balzer for making his serial C++ CCVT code publicly available at https://code.google.com/archive/p/ccvt/. This code makes use of Qhull library (http://www.qhull.org/) to compute Voronoi and Delaunay tessellations.

References

Balzer M., Schlomer T., & Deussen O., 2009, *Capacity constrained point distributions: A variant of lloyd's method*, SIGGRAPH'09, 86, 1

Barber C. B., Dobkin D. P., & Huhdanpaa H. T., 1996, *The Quickhull algorithm for convex hulls*, ACM Transactions on Mathematical Software, 22, 469

Efstathiou G., Davis M., White S. D. M., & Frenk C. S., 1985, *Numerical techniques for large cosmological N-body simulations*, ApJS, 57, 241

Hansen S. H., Agertz O., Joyce M., et al. 2007, *An alternative to grids and glasses: Quaquaversal pre-initial conditions for N-body simulations*, ApJ, 656, 631

Liao S., 2018, *An alternative method to generate pre-initial conditions for cosmological N-body simulations*, arXiv:1807.03574

White S. D. M., 1996, *Formation and evolution of galaxies*, in Schaefer R., Silk J., Spiro M., Zinn-Justin J., eds, Cosmology and Large-Scale Structure, Elsevier, Amsterdam, p. 349

Xu Y., Liu L., Gotsman C., & Gortler S. J., 2011, *Capacity-Constrained Delaunay Triangulation for point distributions*, Computers & Graphics, 35, 510