

# METROC++

*A parallel, object-oriented code for the  
computation of merger trees in cosmological  
simulations*

Edoardo Carlesi  
Leibniz Institut für Astrophysik  
Potsdam, Germany  
ecarlesi@aip.de

November 15, 2018

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Basic usage</b>	<b>2</b>
2.1	Compiling the code . . . . .	2
2.2	Code operation modes . . . . .	2
2.3	Configuration file . . . . .	2
2.4	Temporary files . . . . .	2
<b>3</b>	<b>Examples</b>	<b>3</b>
3.1	Full box simulation . . . . .	3
3.2	Zoom simulation . . . . .	3
<b>4</b>	<b>Editing the code</b>	<b>3</b>
4.1	Other halo finders . . . . .	3
4.2	Improving the algorithms . . . . .	3

# 1 Introduction

METROC++ is an acronym that stands for **MErgerTRees On C++**, and refers to the infamous **Metro C** subway line in Rome, where the author of the code grew up and lived for more than 20 years. This legendary piece of infrastructure (which is still - in 2018 - largely under construction, though a shorter section of it is already operating) took something of the order of a few Giga-Years to be build, a time span which can be compared to the formation time of most dark matter halos.

The code has been written in **C++** and relies on **MPI2.0** C-bindings for the parallelization.

In addition to the source code, a number of **bash** and **python** scripts are provided to ease the

The properties of the algorithms used by the code are described in the code paper (Carlesi, MNRAS, 2018).

## 2 Basic usage

### 2.1 Compiling the code

The main folder contains a `Makefile.config` file

### 2.2 Code operation modes

### 2.3 Configuration file

### 2.4 Temporary files

The code produces a number of temporary files (`.tmp` format extension, located in the `tmp/` folder). These files are needed at runtime to They can be produced manually or using the scripts (`find.z.sh`, `find.n.sh` located in the `scripts/` folder) and contain the list of files on which the halo finder will run on. This can be the full list of snapshots in one simulation, or can be edited to be only a subset of it.

## 3 Examples

### 3.1 Full box simulation

### 3.2 Zoom simulation

## 4 Editing the code

### 4.1 Other halo finders

Although METROC++ was conceived and mainly tested using the **AHF** halo finder, it can be easily extended to support other software as well, as long as:

- Halo catalogues include informations about the number and types of particles, positions and velocities for each object
- Particle catalogs contain the (unique) IDs for each halo particles' content

To add

### 4.2 Improving the algorithms