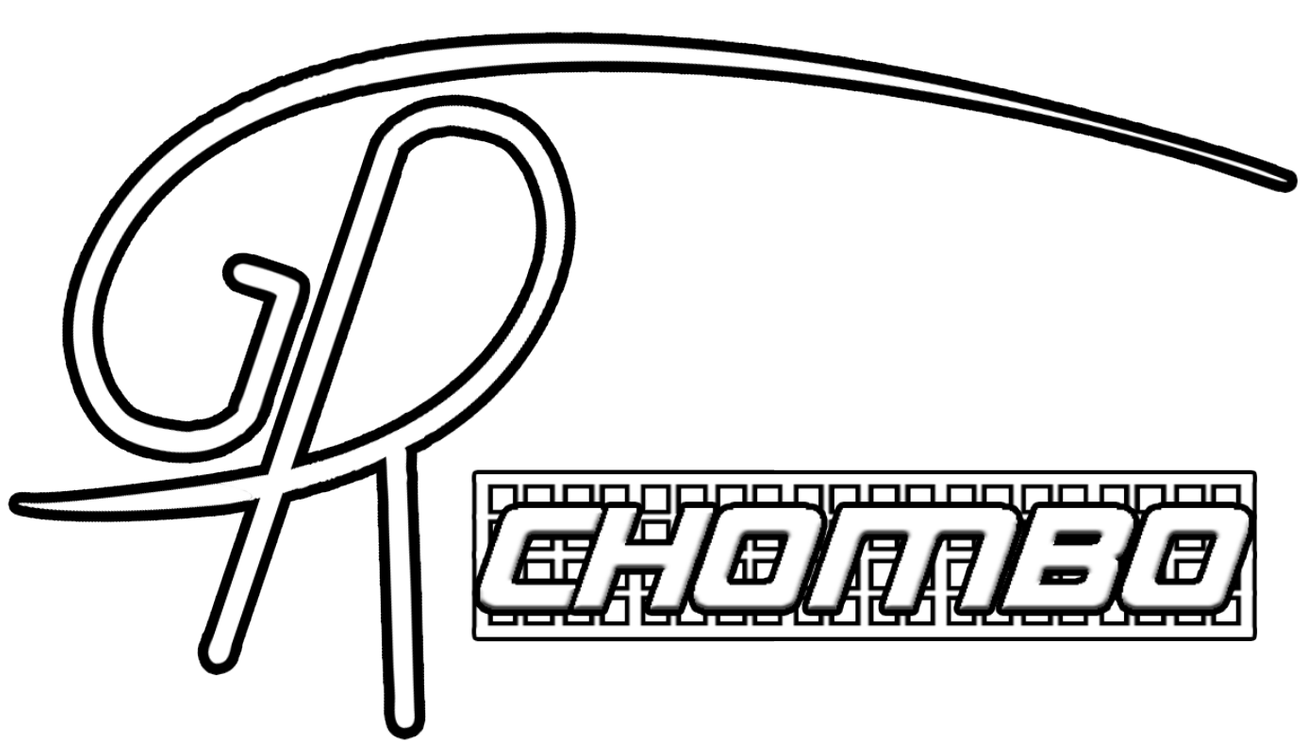
Parameters Guide

Amelia Drew



Includes:

- Parameters in BBH example params file
- Outline parameters in ChomboParameters
- How to add new parameters

Does not include:

- Details about GR parameters

BBH Params Example

```
verbosity = 0
chk_prefix = BinaryBH_
plot_prefix = BinaryBHPlot_
#restart_file = BinaryBH_000360.3d.hdf5
```

Increase for more diagnostic info

Checkpoint from which to restart simulation

```
# Set up grid spacings and regrid params
# NB - the N values need to be multiples of block_factor

N1 = 256
N2 = 256
N3 = 256
L = 192
```

For MPI - see later

N - number of coarsest mesh points for each dimension

L - length of longest box side, dx_coarsest = L/N(max)

Where to regrid e.g. `GRChombo/Source/TaggingCriteria/PhiAndKTaggingCriterion.hpp`

```
data_t criterion = m_dx * (sqrt(mod_d1_phi) / m_threshold_phi +
                           sqrt(mod_d1_K) / m_threshold_K);
```

`GRChombo/Source/GRChomboCore/GRAMRLevel.cpp`

```
IntVect iv(ix, iy, iz);
if (tagging_criterion(iv, 0) >= m_p.regrid_threshold)
{
```

```
regrid_threshold = 0.20
max_level = 5
regrid_interval = 512 256 32 16 8 4 2 2 2
isPeriodic = 1 1 1

#Max and min box sizes
max_grid_size = 32
block_factor = 16
```

Max AMR level (starts from 0)

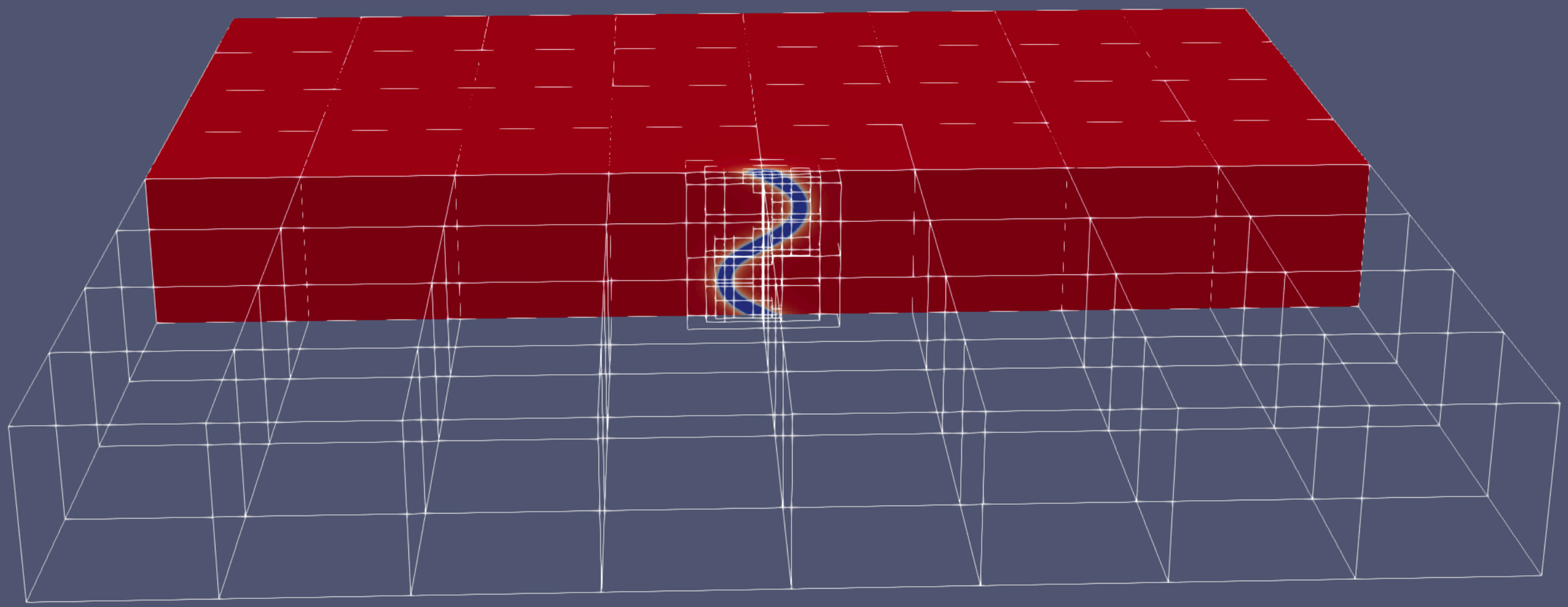
How often to regrid, e.g.:

- 512 - regrid level 1 every 512 coarsest steps
- 256 - regrid level 2 every 256/`ref_ratio` coarsest steps
- 32 - regrid level 3 every 32/(`ref_ratio`^2) coarsest steps etc.

Periodic BCs - instructions in params file (see `BoundaryConditions.hpp` for details)

Max AMR box size

Min AMR box size



```
# Set up time steps
# dt will be dx*dt_multiplier on each grid level
# HDF5files are written every dt = L/N*dt_multiplier*checkpoint_interval

checkpoint_interval = 4
# set to zero to turn off plot files
plot_interval = 0
dt_multiplier = 0.25
stop_time = 1.0
max_steps = 2

nan_check = 1


#coefficient for KO numerical dissipation
sigma = 0.3
```

Self explanatory!

Kreiss-Oliger coefficient for dissipating high frequency modes - max value depends on dt_multiplier

```
#extraction params
#default center to grid center, uncomment to change
#extraction_center = 64 64 64
activate_extraction = 1
num_extraction_radii = 3
extraction_radii = 10. 30. 50.
extraction_levels = 2 1 1
num_points_phi = 16
num_points_theta = 24
num_modes = 3
modes = 2 0 # l m for spherical harmonics
        2 1
        2 2
```

Recent addition - extraction of field values on a surface

Parameters self-explanatory!

Outputs extraction for each timestep in separate text file

```
activate_extraction = 1
write_extraction = 1

write_plot_ghosts = 0
```

Important for visualisation

Adding New Parameters:

1. Create `SimulationParameters.hpp`, inherit from from `SimulationParametersBase` or `ChomboParameters` (no GR)

2. Define parameters and collections of parameters if needed (cleaner passing to functions in ..Level.cpp), e.g. `GRChombo/Examples/AxionString/AxionStringLevel.cpp`

```
BoxLoops::loop(make_compute_pack(SetValue(0.0), AxionString(m_p.initial_params, m_dx, xsect)),
               m_state_new, m_state_new, FILL_GHOST_CELLS, disable_simd());
```

3. Load parameter values from params file

```
#ifndef SIMULATIONPARAMETERS_HPP_
#define SIMULATIONPARAMETERS_HPP_

//General includes
#include "ChomboParameters.hpp"
#include "GRParmParse.hpp"

//Problem specific includes
#include "AxionString.hpp"
#include "CylindricalExtraction.hpp"

class SimulationParameters : public ChomboParameters
{
public:
  SimulationParameters(GRParmParse &pp) : ChomboParameters(pp)
  {
    read_params(pp);
  }

  /// Read parameters from the parameter file
  void read_params(GRParmParse &pp)
  {
```

```
//Collection of parameters necessary for initial conditions
AxionString::params_t initial_params;

//Collection of parameters necessary for extraction
CylindricalExtraction::params_t extraction_params_cylinder;
```

```
// General parameters
int N3;
string path_to_ICs;
int damping;
Real regrid_threshold_phi;
double sigma; // Kreiss-Oliger dissipation parameter
int nan_check;
```

```
// Initial data
pp.load("amplitude", initial_params.amplitude);
pp.load("centerSF", initial_params.centerSF);
//        center; // Default centerSF to center in ChomboParameters

// Cross-section parameters
pp.load("lambda", initial_params.lambda);
pp.load("N3", N3);

initial_params.zlength = N3 * (L / max_N);
```

Note Default Parameters:

- Defined in `GRChombo/Source/GRChomboCore/ChomboParameters.hpp`

- Important: `ref_ratio` defaults to 2 - ratio of dx between refinement levels

```cpp
#ifndef CHOMBOPARAMETERS_HPP_
#define CHOMBOPARAMETERS_HPP_

// General includes
#include "BoundaryConditions.hpp"
#include "GRParmParse.hpp"

class ChomboParameters
{
  public:
    ChomboParameters(GRParmParse &pp) { read_params(pp); }

    void read_params(GRParmParse &pp)
    {
        pp.load("verbosity", verbosity, 0);
        // Grid setup
        pp.load("L", L, 1.0);
        pp.load("center", center,
                {0.5 * L, 0.5 * L, 0.5 * L}); // default to center
        pp.load("regrid_threshold", regrid_threshold, 0.5);
        pp.load("num_ghosts", num_ghosts, 3);
        pp.load("tag_buffer_size", tag_buffer_size, 3);
        pp.load("dt_multiplier", dt_multiplier, 0.25);
        pp.load("fill_ratio", fill_ratio, 0.7);

        // Periodicity and boundaries
        pp.load("isPeriodic", isPeriodic, {true, true, true});
        int bc = BoundaryConditions::STATIC_BC;
        pp.load("hi_boundary", boundary_params.hi_boundary, {bc, bc, bc});
        pp.load("lo_boundary", boundary_params.lo_boundary, {bc, bc, bc});
```

Note: this is only centre for cubic boxes