

bd_top User's Manual

Gary Huber

October 6, 2023

1 Introduction

As in Browndye 1.0, **bd_top** automates some of the preparation for the actual Browndye simulation. It does this by running some of the auxilliary programs and keep track of the intermediate files. For now, it works for two general cases.

First, it can handle an arbitrary number of rigid cores when the molecular mechanics model is being used. In other words, it has the functionality of Browndye 1.0 but with more than two cores possible.

Second, if the COFFDROP model is being used, it can handle an arbitrary number of rigid cores and chains connected into one or two groups. A chain can be unattached to a core, attached at one end, or attached to by two ends to the same core or to two different cores. For now, the simulator cannot handle more than two groups. A *group* is a collection of cores and chains that are all connected and cannot drift apart.

The inputs include, for each core, the XML output of *pqr2xml* and zero or more APBS grids. For each chain, the input includes the XML output of *generate_coffdrop_beads*, and some optional lines that describe how it is attached to the cores.

When *bd_top* runs, it generates several intermediate files for the solvent and each core and chain, and the simulation file that is later given to *nam_simulation* or *we_simulation*. It is possible to make changes to any of the intermediate files, and if *bd_top* is run again, it updates only those files that depend on the one that was changed.

The simulation file that is generated has the form *name1_simulation.xml* or *name1_name2_simulation.xml*, where *name1* and *name2* are the names of the groups.

2 File Structure

```
<root>
  <n_threads> integer </n_threads> # optional
  <seed> integer </seed>
  <output> filename </output>

  # for nam_simulation
  <n_trajectories> integer </n_trajectories>
  <n_trajectories_per_output> integer </n_trajectories_per_output> # optional
  <max_n_steps> integer </max_n_steps>
  <trajectory_file> filename </trajectory_file> # optional
  <n_steps_per_output> integer </n_steps_per_output> # optional
  <min_rxn_dist_file> filename </min_rxn_dist_file> # optional

  # for we_simulation
  <n_copies> integer </n_copies>
  <n_bin_copies> integer </n_bin_copies>
  <n_steps> integer </n_steps>
  <n_we_steps_per_output> integer </n_we_steps_per_output>
  <bin_file> filename </bin_file>

  # system description
  <system>
    <force_field> name </force_field>
    <parameters> filename </parameters>
    <connectivity> filename </connectivity> # needed for COFFDROP
    <start_at_site> true/false </start_at_site> # optional
    <b_radius> real </b_radius> # optional
    <reaction_file> filename </reaction_file> # optional
    <hydrodynamic_interactions> true/false </hydrodynamic_interactions>
    <n_steps_between_hi_updates> integer <n_steps_between_hi_updates> #

    <density_field> filename </density_field> # optional
    <atom_weights> filename </atom_weights> # optional

  <solvent>
```

```

    <debye_length> real </debye_length>
    <dielectric> real </dielectric>
    <relative_viscosity> real </relative_viscosity>
    <kT> real </kT>
    <desolvation_parameter> real </desolvation_parameter>
    <solvent_radius> real </solvent_radius>
</solvent>

<time_step_tolerances> # optional
    <force> real </force>
    <reaction> real </reaction>
    <minimum_core_dt> real </minimum_core_dt>
    <minimum_chain_dt> real </minimum_chain_dt>
    <minimum_core_reaction_dt> real </minimum_core_reaction_dt>
    <minimum_chain_reaction_dt> real </minimum_chain_reaction_dt>
</time_step_tolerances>

<group>
    <name> name </name>
    <core>
        <name> name </name>
        <atoms> filename </atoms>
        <all_in_surface> true/false </all_in_surface> # optional

        <electric_field> # optional
            <grid> filename </grid> # one or more
        </electric_field>

        <dielectric> real </dielectric>

    </core>
    # zero or more

    <chain>
        <name> name </name>
        <atoms> file </atoms>
        <constant_dt> real </constant_dt> # optional
        <constraints> true/false </constraints> # optional
    </chain>

```

```

    <link>
      <core_name> name </core_name>
      <core_residue> integer </core_residue>
      <chain_residue> integer </chain_residue>
    </link>
    # 0, 1, or 2

  </chain> # zero or more

</group>
# 1 or 2

</system>

</root>

```

Most of the XML tags are identical to those in the simulation file, and are simply copied over. The exceptions are noted below.

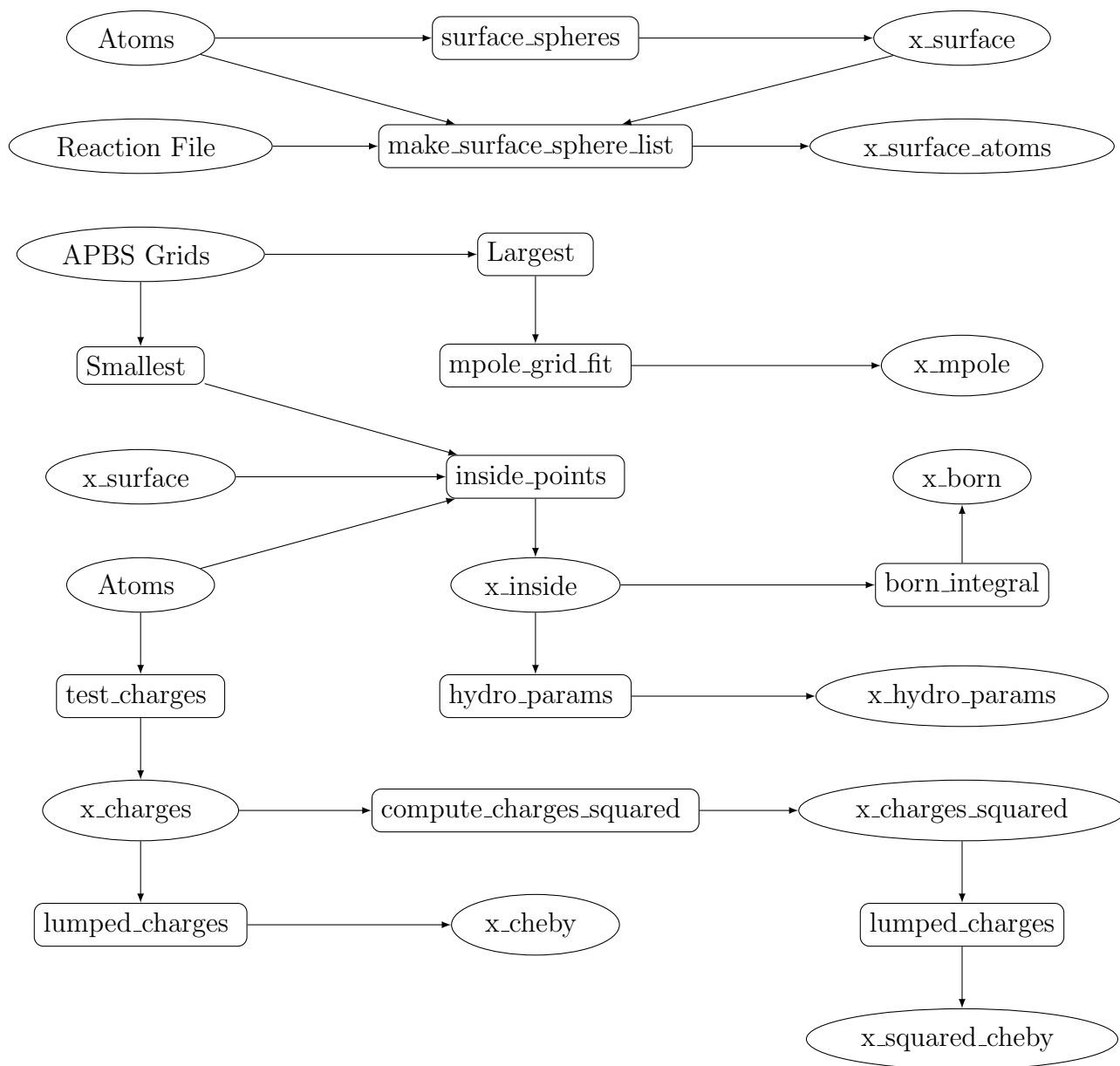
- **solvent** - The contents are simply copied over to a newly generated solvent file, which is an input to the simulation program. One addition is the **solvent_radius** tag, which represents the probe radius and exclusion radius in various auxilliary programs. If left out, the default value is 1.5 Å.
- **connectivity** - this file describes how the beads in the chain model are connected. For COFFDROP, the file *connectivity.xml* from the website can be used.
- **link** - This sits within the **chain** tag. Each instance of this item describes a connection between the chain and a core. The name of the core, the number of the linking residue, and the number of the chain's linking residue are specified.
- **all_in_surface** - If this is true, then all of the atoms of the core are included, not just a shell made of the surface atoms. Default is false.
- **is_protein** - If this is true, then it affects how the effective charges are generated. Default is false.

- **atoms** - When using the COFFDROP model, the atoms files for cores and chains must contain the coarse-grained “beads”. This will usually be the output of *generate_coffdrop_beads*.
- **constraints** - If true, all of the bonds are treated as rigid constraints. Default is false.

3 Programs and Files

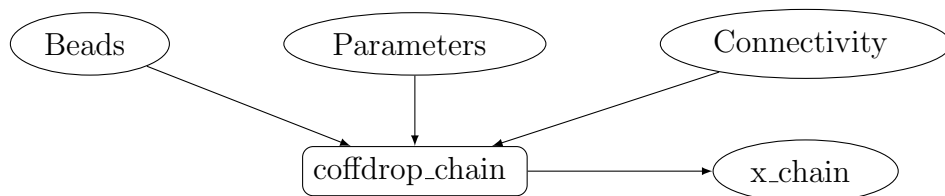
3.1 For Each Core

This is very similar to the flow in Browndye 1. The files are denoted by ovals, and it is assumed that the core name is “x”. The file names that result are shown without the .xml file extension. Several of the programs take solvent parameters from the input file. Although those dependencies are not shown in the diagram, if a solvent parameter changes, the programs taking it as an input are rerun upon rerunning *bd_top*.



3.2 For Each Chain

Here, we assume that the name of the chain is “x”.



4 Final Notes

When setting up a COFFDROP simulation with rigid cores, the APBS files must be generated from the output of *generate_coffdrop_beads*, and the dielectric of the molecule must be the same as the dielectric of the solvent. This is because the dielectric boundary effects are presumed to be accounted for in the COFFDROP potential.

There is also a program, **scale_interactions**, that can be used to scale the nonbonded attractive COFFDROP interactions per the examples in the original COFFDROP papers.