Fermion Bag Hamiltonian Lattice Field Theory Code Manual

Joseph Buck
Emilie HuffmanDepartment of Physics, Wake Forest University, Winston-Salem, NC

August 15, 2025

Contents

1 Introduction

2 Theoretical Background

3 Software Overview

4 Installation and Setup

5 Input Parameters

5.1 Input File Format

The input files for this code are YAML files, so parameter specifications are in the form of key-value pairs. Variables are often nested under a few keys for increased clarity. There are four primary categories of variables that can be set in these input files: control, lattice, configuration, and constants.

5.2 Input Parameters

- `control` Control parameters are any variables determining the overall execution of the code.
  - `algorithm` The bond switching algorithm to use in the relaxation process. Currently, the only option is "random", but others will be introduced as development proceeds.
    - `random` This algorithm takes a starting configuration and randomly removes a bond from the configuration, randomly chooses a new tau, and randomly (using the bond type proportions) chooses the new bond size and lattice sites. This proceeds for nmoves iterations. For more information, see 7.
  - `nmoves` The number of times the configuration should be updated.
- `lattice` Lattice parameters include all variables needed to fully determine the lattice structure.
  - `type` The Bravais lattice type. The only supported type is "simple-cubic" in one dimension at the moment. Support will be added for up to three dimensions and a variety of Bravais lattice types.
  - `a, b, c` The axial lengths of the Bravais lattice in Bohr. If only a is supplied, the lattice will be one dimensional. If a and b are supplied it will be two, and if all three are supplied it will be three dimensional.

- `alpha, beta, gamma` The angles completing the Bravais lattice definition in radians. If only a is supplied, these angles will be ignored. If a and b are supplied, only alpha is required. If a, b, and c are supplied, all angles are required.
- `lims` The lims section is for defining information on the overall spatial extent of the lattice.
  - `x, y, z` Variables x, y, and x are subcategories of the lims section. If a is supplied, x must be set, etc.
    - `min` The minimum value in Bohr that the lattice should extend to in the given dimension.
    - `max_factor` Instead of setting a maximum limit for the given dimension, you choose the maximum number of multiples of the unit cell as a way of specifying the upper limit. This parameter, along with "base", determines the upper limit of the lattice.
    - `base` The multiplier which acts as the "unit" that max_factor is multiplied by to determine the upper boundary.

- `configuration` Configuration parameters fully determine the attributes of configurations throughout the development of the code. For definitions of configurations and related concepts, see Sections 2 and 7.
  - `float_tol` The float tolerance determines how precise the tau variable needs to be when retrieving the bond at a given tau. Bonds are stored in a hashmap, mapping from the tau float variable to a bond object. If no floating point tolerance is specified, machine precision discrepancies could contribute to the code not recognizing a given tau. The default float tolerance is 1e-5.
  - `nbonds` The total number of bonds to be included in a configuration. Note that the actual number of bonds in a given configuration may be slightly less than nbonds in order to accomodate the bond type proportions (bond_type_props). The actual number of bonds will never be greater than nbonds. This variable by definition also determines the number of unique taus in a configuration.
  - `tau_max` The tau variable in the code is an "imaginary" time variable. This tau_max input parameter determines the maximum allowed value that tau is allowed to take.
  - `bond_type_props` Bonds in the code can be made up of one or more adjacent lattice sites. The purpose of this parameter is to specify which bond sizes should be included in the simulation, and

the relative proportions of each size. Bond sizes are given as keys under the bond_type_props section. The relative proportions are given as the values corresponding to each key. The proportions can be given as floats or integers. Renormalization and computation of the actual number of each type of bond is done in the code in conjunction with nbonds.

- `constants` This parameter section is the place to define useful constants that may be needed to fully specify the input. For example, Bravais angles for a simple-cubic lattice need to be set to pi/2. So, pi/2 would be a useful constant to store here. The code does not rely on these user constants to perform its calculations. Where those are needed, they are defined in the code.

5.3 Environment Variables

6 Output Files and Data

7 Physics Implementations

8 Extending and Contributing to the Code

9 Testing and Validation

10 Examples and Tutorials

11 Appendices

References