PyCD Documentation

PyCD

Apr 29, 2020

Contents:

CHAPTER 1

Getting Started

This page details how to get started with PyCD.

CHAPTER 2

API Documentation

class PyCD.**Material**(*material_parameters*)

Defines the properties and structure of working material

generate_sites(*element_type_indices*, *cell_size*)

Generates NumPy array of sites for the specified element types and cell size

Parameters

- **element_type_indices**
- **cell_size** (*np.ndarray*) – size of the cell

Returns **return_sites** – Object of cell_coordinates, quantum_index_list, system_element_index_list

Return type object

class PyCD.**Neighbors**(*material*, *system_size*, *pbc*)

Returns the neighbor list file :param system_size: size of the super cell in terms of number of

unit cell in three dimensions

compute_distance(*system_size*, *system_element_index_1*, *system_element_index_2*)

Returns the distance in atomic units between the two system element indices for a given system size :param system_size: :param system_element_index_1: :param system_element_index_2: :return:

generate_neighbor_list(*dst_path*, *local_system_size*)

Adds the neighbor list to the system object and returns the neighbor list :param dst_path: :param local_system_size: :return:

get_coordinates(*system_size*, *system_element_index*)

Returns the coordinates in atomic units of the given system element index for a given system size :param system_size: :param system_element_index: :return:

get_pairwise_min_image_vector_data(*dst_path*)

Returns cumulative displacement list for the given system size printed out to disk :param dst_path: :return:

get_quantum_indices(*system_size*, *system_element_index*)
Returns the quantum indices of the element :param system_size: :param system_element_index: :return:

get_system_element_index(*system_size*, *quantum_indices*)
Returns the system_element_index of the element :param system_size: :param quantum_indices: :return:

hop_neighbor_sites(*bulk_sites*, *center_site_indices*, *neighbor_site_indices*, *cutoff_dist_limits*, *cutoff_dist_key*)
Returns system_element_index_map and distances between center sites and its neighbor sites within cutoff distance :param bulk_sites: :param center_site_indices: :param neighbor_site_indices: :param cutoff_dist_limits: :param cutoff_dist_key: :return:

class PyCD.**System**(*material_info*, *material_neighbors*, *hop_neighbor_list*, *pairwise_min_image_vector_data*, *alpha*, *r_cut*, *k_cut*, *precision_parameters*, *step_system_size_array*, *step_hop_neighbor_master_list*)
defines the system we are working on

Attributes: size: An array (3 x 1) defining the system size in multiple of unit cells

get_precomputed_array(*dst_path*, *compute_energy_contributions*, *return_k_vector_data*)

Parameters **dst_path** –

Returns

pot_k_ewald(*k_max*, *alpha*, *k_cut*)
Updates precomputed array with potential energy contributions from reciprocal-space

pot_k_ewald_with_k_vector_data(*charge_list_prod*, *k_max*, *alpha*, *k_cut*)
Updates precomputed array with potential energy contributions from reciprocal-space

pot_r_ewald(*alpha*, *r_cut*)
Generates precomputed array with potential energy contributions from real-space confined to simulation cell i.e. n_max=[0, 0, 0]

class PyCD.**Run**(*system*, *precomputed_array*, *temp*, *ion_charge_type*, *species_charge_type*, *n_traj*, *t_final*, *time_interval*, *species_count*, *initial_occupancy*, *relative_energies*, *external_field*, *doping*)
defines the subroutines for running Kinetic Monte Carlo and computing electrostatic interaction energies

charge_config(*occupancy*, *dopant_site_indices*)
Returns charge distribution of the current configuration :param occupancy: :param ion_charge_type: :param species_charge_type: :return:

do_kmc_steps(*dst_path*, *output_data*, *random_seed*, *compute_mode*)
Subroutine to run the KMC simulation by specified number of steps :param dst_path: :return:

generate_initial_occupancy(*dopant_site_indices*)
generates initial occupancy list based on species count :param species_count: :return:

preproduction(*dst_path*, *random_seed*)
Subroutine to setup input files to run the production stage of the simulation :param dst_path: :param random_seed: :return:

class PyCD.**Analysis**(*material_info*, *n_dim*, *species_count*, *n_traj*, *t_final*, *time_interval*, *msd_t_final*, *trim_length*, *temp*, *repr_time='ns'*, *repr_dist='Angstrom'*)
Post-simulation analysis methods

compute_msd(*dst_path*, *output_data*)
Returns the squared displacement of the trajectories :param dst_path: :return:

generate_msd_plot(*msd_data*, *sem_data*, *display_error_bars*, *species_types*, *file_name*, *dst_path*)
Returns a line plot of the MSD data :param msd_data: :param std_data: :param display_error_bars: :param species_types: :param file_name: :param dst_path: :return:

class PyCD.material_setup.**ReturnValues**(*input_dict*)
dummy class to return objects from methods defined inside other classes

PyCD.material_setup.**material_setup**(*input_directory_path*, *system_size*, *pbc*, *generate_hop_neighbor_list*, *generate_pairwise_min_image_vector_data*, *generate_precomputed_array*, *compute_energy_contributions*, *return_k_vector_data*)
Prepare material class object file, neighbor list and saves to the provided destination path

class PyCD.material_preprod.**ReturnValues**(*input_dict*)
dummy class to return objects from methods defined inside other classes

class PyCD.material_run.**ReturnValues**(*input_dict*)
dummy class to return objects from methods defined inside other classes

class PyCD.material_msd.**ReturnValues**(*input_dict*)
dummy class to return objects from methods defined inside other classes

PyCD.io.**generate_report**(*start_time*, *dst_path*, *file_name*, *print_time_elapsed*, *prefix=None*)
Generates a report file to the output directory This function writes out a VASP-style POSCAR file and returns dict containing a variety of POSCAR-derived quantities.

Parameters

- **start_time** (*datetime.datetime*) – Time value when this function is called
- **dst_path** (*str*) – The location of the destination file to write out
- **file_name** (*str*) – The file name of the output report
- **print_time_elapsed** (*bool*) – Flag value to whether to print elapsed time or not
- **prefix** (*str or NoneType*) – String value to prefix the newly generated report content

Returns None

Return type NoneType

PyCD.io.**read_poscar**(*input_file_path*)
This function reads in a VASP-style POSCAR file and returns dict containing a variety of POSCAR-derived quantities.

Parameters **file_location** (*str*) – The location of the POSCAR file to read in

Returns **poscar_info** – Dict of POSCAR-derived quantities

Return type dict of {str : np.ndarray, str : list, str : list, str : int, str : str, str : np.ndarray, str : str}

PyCD.io.**write_poscar**(*src_file_path*, *dst_file_path*, *file_format*, *element_types_cluster*, *num_elements_cluster*, *coordinate_type*, *coordinates_cluster*)
This function writes out a VASP-style POSCAR file and returns dict containing a variety of POSCAR-derived quantities.

Parameters

- **src_file_path** (*str*) – The location of the source file to read in
- **dst_file_path** (*str*) – The location of the destination file to write out
- **file_format** (*str*) – The file format of the source coordinate file: VASP, VESTA, unknown
- **element_types_cluster** (*list*) – List of element types in the cluster
- **num_elements_cluster** (*list*) – List containing number of elements for each element type listed in element_types_cluster

- **coordinate_type** (*str*) – Nature of coordinates : Cartesian or Fractional
- **coordinates_cluster** (*np.ndarray*) – NumPy array containing the coordinates within the selected cluster

Returns **None**

Return type NoneType

class PyCD.scripts.data_profile.**DataProfile**(*dst_path*, *system_directory_path*, *variable_quantity_type_index*, *variable_quantity_index*, *variable_quantity_list*, *species_count*, *t_final*, *time_interval*, *n_traj*, *external_field*, *doping*)

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index