PyFaults Documentation
Author: Sinclair R. Combs

User Input

Layered unit cell
The program requires *a priori* identification of layers within the unit cell. These should be loaded into the program as a csv file of the following format, where each cell represents a csv entry:

Layer	Atom	Element	x	y	z	Occupancy
A	Sc1	Sc^{3+}	0	0	0	1
A	Li1	Li^{+}	0	0.3347	0	1
A	Li2	Li^{+}	0.5	0.169	0	0.365
B	Sc1	Sc^{3+}	0	0.5	0.5	1
B	Li1	Li^{+}	0	0.1653	0.5	1
B	Li2	Li^{+}	0	0.331	0.5	0.365

Layer is a unique tag associated with a given layer of atoms. **Atom** is a unique tag to distinguish from other atoms of the same type. **Element** is the atomic element and oxidation state. The parameters **x, y, and z** are the atomic position in fractional coordinates. **Occupancy** is the site occupancy.
Note: Unit cell should be transformed (if necessary) such that layers are orthogonal to lattice vectors. When importing the csv, the stacking direction will automatically be converted to c.

Supercell size (N) – Number of repeat unit cells used to construct supercells

Stacking direction – Lattice vector orthogonal to layers

Faulted layer – Unique tag of layer to be faulted

Stacking probability (P) – Probability with which faulting will occur in a given layer

Stacking vector (S) – Translation vector corresponding to stacking fault displacement in fractional coordinates relative to the original layer position

Package API

pyfaults

Top-level module.

Functions

`toCif` (*cell, path, fn, inclPkl=False*)
Generates a CIF file from a given structure.

Parameters	cell (*Unitcell or Supercell*) – Structure to write to cif format path (*str*) – File directory to save to fn (*str*) – File name to save to inclPkl (*bool, optional*) – Generates pickle file of cif data. Default is False.
Returns	None

`importCSV` (*path, fn*)
Imports CSV file as a dataframe.

Parameters	path (*str*) – File directory fn (*str*) – File name
Returns	None

Classes

`Lattice`
: Lattice parameters

`LayerAtom`
: Atom in a specified layer and its properties

`Layer`
: Collection of atoms that form a specified layer

`ChildLayer`
: Spatially translated child layer from parent layer

`Unitcell`
: Collection of layers that form a unit cell

`Supercell`
: Collection of unit cells that form a supercell

pyfaults.lattice

Contains class Lattice for defining lattice parameters.

class `Lattice` (*a, b, c, alpha, beta, gamma*)
Bases: `object`
Stores lattice parameter values.

Parameters	a (*float*) – Lattice vector a (Å) b (*float*) – Lattice vector b (Å) c (*float*) – Lattice vector c (Å) alpha (*float*) – Lattice angle α (°) beta (*float*) – Lattice angle β (°) gamma (*float*) – Lattice angle γ (°)

Functions

`setParam` (*a=None, b=None, c=None, alpha=None, beta=None, gamma=None*)
Sets one or more lattice parameters, any parameters not specified will retain original values.

Parameters	a (*float, optional*) – New value of lattice vector a (Å) b (*float, optional*) – New value of lattice vector b (Å) c (*float, optional*) – New value of lattice vector c (Å) alpha (*float, optional*) – New value of lattice angle α (°) beta (*float, optional*) – New value of lattice angle β (°) gamma (*float, optional*) – New value of lattice angle γ (°)
Returns	*Self*

`display` ()
Prints lattice parameters.

Parameters	None
Returns	None

pyfaults.layerAtom

Contains class LayerAtom for defining atoms in specific layers.

class `LayerAtom` (*layerName, atomLabel, element, xyz, occupancy, lattice*)
Bases: `object`
Stores layer-specific atomic properties.

Parameters	layerName (*str*) – Unique identifier for layer in which atom is located atomLabel (*str*) – Unique identifier for atom element (*str*) – Atomic element and oxidation state (if applicable) xyz (*nparray*) – Atomic position [x, y, z] in fractional coordinates occupancy (*float*) – Crystallographic site occupancy lattice (*Lattice*) – Unit cell lattice parameters

Functions
`setParam` (*layerName=None, atomLabel=None, element=None, xyz=None, lattice=None, occupancy=None*)
Sets one or more atomic parameters, any parameters not specified will retain original values.

Parameters	layerName (*str, optional*) – New layer name atomLabel (*str, optional*) – New atom label element (*str, optional*) – New element label xyz (*nparray, optional*) – New array for atomic position occupancy (*float, optional*) – New value for site occupancy lattice (*Lattice, optional*) – New unit cell lattice
Returns	*Self*

`display` ()
Prints atomic parameters.

Parameters	None
Returns	None

pyfaults.layer

Contains class Layer for storing a list of atoms that defines a layer.

class `Layer` (*atoms, lattice, layerName*)
Bases: `object`
Collection of atoms that form a layer.

Parameters	atoms (*LayerAtom*) – List of LayerAtom instances to add to layer lattice (*Lattice*) – Unit cell lattice parameters layerName (*str*) – Unique identifier for layer

Functions
`setParam` (*atoms=None, lattice=None, layerName=None*)
Sets one or more layer parameters, any parameters not specified will retain original values.

Parameters	atoms (*LayerAtom, optional*) – New list of atoms to add to layer lattice (*Lattice, optional*) – New unit cell lattice layerName (*str, optional*) – New layer name
Returns	*Self*

`display` ()
Prints layer name and atoms.

Parameters	None
Returns	None

`atom_info` ()
Prints atomic information of atoms in layer.

Parameters	None
Returns	None

Additional Methods
`getLayers` (*df, lattice, layerNames, stackDir*)
Pulls layer information from imported dataframe.

Parameters	df (*dataframe*) – Dataframe with atomic parameters lattice (*Lattice*) – Unit cell lattice parameters layerNames (*str*) – Layer names as documented in imported csv stackDir (*str*) – Lattice vector orthogonal to layers
Returns	layers (*Layer*) – List of layer instances

pyfaults.childLayer

Contains class ChildLayer for generating a spatially translated copy of a parent layer.

class `ChildLayer` (*layerName, parent, transVec*)
Bases: `object`
Generates a child layer from a given parent layer.

Parameters	layerName (*str*) – Unique identifier for child layer parent (*Layer*) – Parent layer transVec (*nparray*) – Translation vector [x, y, z] representing shift from parent layer position to child layer position in fractional coordinates

Functions
`setParam` (*layerName=None, parent=None, transVec=None*)
Sets one or more child layer parameters, any parameters not specified will retain original values.

Parameters	layerName (*str, optional*) – New child layer name parent (*Layer, optional*) – New parent layer transVec (*nparray, optional*) – New translation vector
Returns	*Self*

`generate` (*parent, transVec*)
Creates child layer by translating parent layer.

Parameters	parent (*Layer*) – Parent layer transVec (*nparray*) – Translation vector [x, y, z] representing shift from parent layer position to child layer position in fractional coordinates
Returns	*Self*

`display` ()
Prints layer name and atoms.

Parameters	None
Returns	None

`atom_info` ()
Prints atomic information of atoms in layer.

Parameters	None
Returns	None

pyfaults.unitcell

Contains class Unitcell for storing a list of layers that defines a unit cell.

class `Unitcell` (*name, layers, lattice*)
Bases: `object`
Collection of layers that form a unit cell.

Parameters	name (*str*) – Unique identifier for unit cell layers (*Layer*) – List of layers that form unit cell lattice (*Lattice*) – Unit cell lattice parameters

Functions
`setParam` (*name=None, layers=None, lattice=None*)
Sets one or more unit cell parameters, any parameters not specified will retain original values.

Parameters	name (*str, optional*) – New unit cell name layers (*Layer, optional*) – New list of layers lattice (*Lattice, optional*) – New unit cell lattice
Returns	*Self*

`layer_info` ()
Prints names of layers in unit cell.

Parameters	None
Returns	None

`atom_info` ()
Prints atomic information of atoms in layer.

Parameters	None
Returns	None

pyfaults.Supercell

Contains class Supercell for storing a list of unit cells that defines a supercell.

class `Supercell` (*unitcell, nStacks, fltLayer=None, stackVec=None, stackProb=None*)
Bases: `object`
Collection of unit cells that form a supercell.

Parameters	unitcell (*Unitcell*) – Base unit cell nStacks (*int*) – List of layers that form unit cell fltLayer (*str, optional*) – Name of faulted layer. Default is None. stackVec (*nparray, optional*) – Stacking vector relative to ideal position. Default is None. stackProb (*float, optional*) – Stacking probability. Default is None.

Functions

`setParam` (*nStacks=None*)
Sets value for number of stacks (N).

Parameters	nStacks (*int, optional*) – New N value
Returns	*Self*

`setLayers` (*unitcell, fltLayer=None, stackVec=None, stackProb=None*)
Sets supercell layers, if only unitcell is specified, the supercell will not be faulted.

Parameters	unitcell (*Unitcell*) – Base unit cell fltLayer (*str, optional*) – New name of faulted layer stackVec (*nparray, optional*) – New stacking vector stackProb (*float, optional*) – New stacking probability
Returns	*Self*

`layer_info` ()
Prints names of layers in unit cell.

Parameters	None
Returns	None

`show_faults` ()
Prints names of faulted layers.

Parameters	None
Returns	None