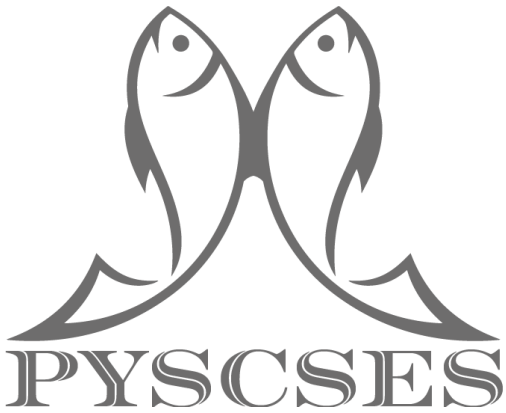
pyscses Documentation

Release 1.0.0

Georgina Wellock

Aug 05, 2019

API documentation



`pyscses` is a Python module that implements a site-explicit, one-dimensional Poisson-Boltzmann solver, used for modelling ionic space charge properties in solid materials. Space charge properties such as electrostatic potential, charge density and charge carrier distributions over the space charge region can be calculated using the Poisson-Boltzmann equation from the input of defect segregation energies and atomically resolved charge carrier positions. The grain boundary resistivity and activation energy can be calculated by extending the model using the calculated charge carrier distributions. `pyscses` also accounts for different approximations typically assumed when space charge formation is considered. These approximations include site explicit vs. continuum modelling, Mott-Schottky (single mobile defect species) and Gouy-Chapman (all defect species mobile) conditions, and whether the charge of the non-defective species should be considered. Full mathematical derivations, definitions and example code can be found in the userguide.

API documentation can be found *here*:

Source code is available as a git repository at https://github.com/bjmorgan/pyscses/tree/master/pyscses.

CHAPTER 1

Installation

The simplest way to install `pyscses` is to use `pip` to install from PyPI:

```
pip install pyscses
```

Alternatively, you can download the latest release from GitHub, and install directly:

```
cd pyscses
pip install -e .
```

which installs an editable (-e) version of pyscses in your userspace.

Or clone the latest version from *GitHub <https://github.com/bjmorgan/pyscses/releases>* with:

```
git clone git@github.com:bjmorgan/pyscses.git
```

and install the same way:

```
cd pyscses
pip install -e .
```

CHAPTER 2

Tests

Jupyter notebooks that can be used to check the output of the calculations can be found in:

pyscses/tests/test_notebooks

The test notebooks can be found on github here.

There are four directories with varying conditions. In each there is a Jupyter notebook which can be run. The input for the calculations is stored in the `input_data` directory and the output for the calculations will be stored in the `generated_data` directory and can be compared to the verified data in the `expected_outputs` directory. A list of the input parameters used in the notebooks is reiterated in each of the four test folders in the `input_parameters` file.

CHAPTER 3

Documentation

Once installed, the `pyscses` code is imported into, and run using a Jupyter notebook. An overview of the capabilities of `pyscses`, with example code for running the code and varying the simulation conditions can be found in:

```
pyscses/userguides/userguide.ipynb
```

or the Jupyter notebook can be found on GitHub here.

API documentation is available here.

CHAPTER 4

Scientific context

In polycrystalline solid materials, grain boundaries and interfaces exist separating different crystalline domains. The structural distortion at these interfaces causes segregation of charge carriers to, or away from the grain boundary core. Due to this, the grain boundary core carries a net charge which causes the depletion or accumulation of charge carriers in the regions adjacent, known as space charge regions. Due to the variation on charge carrier concentrations, the ionic conductivity of the material can be strongly affected by the presence of grain boundaries.

CHAPTER 5

Citing pyscses

This code can be cited as:

Wellock, Georgina L., & Morgan, Benjamin J. (2019). *pyscses - a PYthon Space-Charge Site-Explicit Solver* Zenodo. http://doi.org/10.5281/zenodo.2536867

BibTeX:

```
@misc{wellock_georgina_l_2019_2536901,
  author       = {Wellock, Georgina L. and
                  Morgan, Benjamin J.},
  title        = {{pyscses - a PYthon Space-Charge Site-Explicit
                   Solver}},
  month        = jan,
  year         = 2019,
  doi          = {10.5281/zenodo.2536901},
  url          = {https://doi.org/10.5281/zenodo.2536867}
}
```

5.1 pyscses

5.1.1 pyscses package

Submodules

pyscses.calculation module

class pyscses.calculation.**Calculation**(*grid*, *bulk_x_min*, *bulk_x_max*, *alpha*, *convergence*, *dielectric*, *temp*, *boundary_conditions*)

The Calculation class contains methods for calculating the relevant space charge properties for any given system, such as electrostatic potential, charge density, defect mole fractions and parallel and perpendicular grain boundary resistivities.

Parameters

- **grid** (pyscses.Grid) – A pyscses.Grid object. This contains properties of the grid including the x coordinates and the volumes.
- **bulk_x_min** (*float*) – The minimum x coordinate defining a region of bulk.
- **bulk_x_max** (*float*) – The maximum x coordinate defining a region of bulk.
- **alpha** (*float*) – A damping parameter for updating the potential at each iteration.
- **convergence** (*float*) – The convergence limit. The difference between the updated phi and the phi from the previous iteration must be below this for the calculation to be sufficently converged.
- **dielectric** (*float*) – The dielectric constant for the studied material.
- **temp** (*float*) – The temperature that the calculation is run.
- **boundary_conditions** (*str*) – Specified boundary conditions for the matrix solver. Allowed values are *dirichlet* and *periodic*. Default = *dirichlet*.

calculate_average(*grid*, *min_cut_off*, *max_cut_off*, *sc_property*)

Calculate the average of a given space chage property over a given region.

Parameters

- **grid** (pyscses.Grid) – Contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **min_cut_off** (*float*) – Minimum x coordinate value defining the calculation region.
- **max_cut_off** (*float*) – Maximum x coordinate value defining the calculation region.
- **sc_property** (*list*) – Value of space charge property at all sites.

Returns The average value for the property over the given sites.

Return type float

calculate_debye_length()

Calculate the Debye length.

The output is stored as a Calculation attribute. Calculation.debye_length (float): The Debye length as derived from Poisson-Boltzmann equation.

Parameters **None** –

calculate_delta_x(*grid*, *min_cut_off*, *max_cut_off*)

Calculates the distance between the midpoints of each consecutive site. Inserts the calculated distance to the next grid point outside of the calculation region to the first and last position as the delta_x value for the endmost sites.

Parameters

- **grid** (pyscses.Grid) – Contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **min_cut_off** (*float*) – Minimum x coordinate value defining the calculation region.
- **max_cut_off** (*float*) – Maximum x coordinate value defining the calculation region.

Returns Distance between consecutive sites.

Return type list

calculate_mobile_defect_conductivities(*pos_or_neg_scr*, *scr_limit*, *species*, *mobility_scaling=False*)

Calculate the conductivity ratio between the space charge region and the bulk both perpendicular and parallel to the grain boundary.

A *Set_of_Sites* object is created for the sites in the space charge region, and the defect distributions calculated. The width of the space charge region is calculated and a bulk region of the same width is defined. A Set_of_Sites object for the bulk region is created and the defect distributions calculated. Taking each site as a resistor in series or parallel respectively, the conductivity is calculated and the ratio between the space charge region and the bulk is taken.

Parameters

- **pos_or_neg_scr** (*str*) – 'positive' - for a positive space charge potential. 'negative' - for a negative space charge potential.
- **scr_limit** (*float*) – The minimum electrostatic potential that the electrostatic potential must exceed to be included in the space charge region.
- **species** (*str*) – The species for which the conductivity is being calculated.
- **mobility_scaling** (*bool*) – For particles on a lattice which only interact through volume exclusion, the mobility exhibits a blocking term. True if the blocking term is to be included, False if the blocking term is not to be included. Default = False.

Returns The perpendicular conductivity ratio. The conductivity ratio between the bulk and the space charge region perpendicular to the grain boundary. float: The parallel conductivity ratio. The conductivity ratio between the bulk and the space charge region parallel to the grain boundary.

Return type float

calculate_offset(*grid*, *min_cut_off*, *max_cut_off*)

Calculate the offset between the midpoint of the last x coordinate in the calculation region and the x coordinate outside of the calulation region.

Parameters

- **grid** (pyscses.Grid) – Contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **min_cut_off** (*float*) – Minimum x coordinate value defining the calculation region.
- **max_cut_off** (*float*) – Maximum x coordinate value defining the calculation region.

Returns Offset for the minimum x coordinate. float: Offset fot the maximum x coordinate.

Return type float

calculate_resistivity_ratio(*pos_or_neg_scr*, *scr_limit*, *mobility_scaling=False*)

Each species present in the system is looped over and the perpendicular and parallel conductivity ratios are calculated and appended to separate lists. Once all species have been added to the list, the conductivities are summed and the reciprocal taken to give the resistivity ratios perpendicular and parallel to the grain boundary. The outputs are stored as calculation attributes, Calculation.perpendicular_resistivity_ratio(float), Calculation.parallel_resistivity_ratio (float): $r_{\mathrm{gb}}^{\perp}, r_{\mathrm{gb}}^{\parallel}$: The perpendicular and parallel grain boundary resistivity ratios. :param pos_or_neg_scr: 'positive' - for a positive space charge potential.

'negative' - for a negative space charge potential.

Parameters

- **scr_limit** (*float*) – The minimum electrostatic potential that the electrostatic potential must exceed to be included in the space charge region.

- **mobility_scaling** (*bool*) – For particles on a lattice which only interact through volume exclusion, the mobility exhibits a blocking term. True if the blocking term is to be included, False if the blocking term is not to be included. Default = False.

calculate_space_charge_width(*valence*)

Calculate the approximate space charge width from the Debye length. The output is stores as a Calculation attribute. Calculation.space_charge_width (float): The approximate space charge width.

Parameters **valence** (*float*) – The charge of the mobile defect species.

create_space_charge_region(*grid*, *pos_or_neg_scr*, *scr_limit*)

Calculate the space charge region. The space charge region is defined as the region when the electrostatic potential is greater than a predefined limit.

Parameters

- **grid** (pyscses.Grid) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **pos_or_neg_scr** (*str*) – 'positive' - for a positive space charge potential. 'negative' - for a negative space charge potential.
- **scr_limit** (*float*) – The minimum electrostatic potential that the electrostatic potential must exceed to be included in the space charge region.

Returns List of x coordinates for sites within the space charge region.

Return type list

create_subregion_sites(*grid*, *min_cut_off*, *max_cut_off*)

Creates a *pyscses.Set_of_Sites* object for a defined region of the grid.

Parameters

- **grid** (*object*) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **min_cut_off** (*float*) – Minimum x coordinate value defining the calculation region.
- **max_cut_off** (*float*) – Maximum x coordinate value defining the calculation region.

Returns Set of Sites object for a given subregion of the grid.

Return type pyscses.SetOfSites

find_index(*grid*, *min_cut_off*, *max_cut_off*)

Calculates the indices of the grid positions closest to a minimum and maximum value.

Parameters

- **grid** (pyscses.Grid) – pyscses.Grid object. This contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **min_cut_off** (*float*) – Minimum x coordinate value defining the calculation region.
- **max_cut_off** (*float*) – Maximum x coordinate value defining the calculation region.

Returns Index for minimum cutoff; index for maximum cutoff

Return type int, int

form_subgrids(*site_labels*)

Creates a *pysces.Grid* object for each species in the system. The output is a dictionary of separate Grid classes for the different site species and is stored as Calculation.subgrids.

Parameters **site_labels** (*list*) – List of strings for the different site species.

mole_fraction_correction(*target_mole_fractions*, *approximation*, *initial_guess*)
Starting from an initial guess for the appropriate input mole fractions, minimises the error between the target bulk mole fraction and the output mole fraction from the iterative Poisson-Boltzmann solver. The output is stored as a Calculation attribute. Calculation.initial_guess (list): The optimum values to be used as the input mole fractions for the iterative Poisson-Boltzmann solver so that the output bulk mole fractions are the target bulk mole fractions.

Parameters

- **target_mole_fractions** (*list*) – The value that the mole fractions should be in the bulk.
- **approximation** (*str*) – The defect mobility approximation. Either 'mott-schottky' to enforce only a single mobile defect, or 'gouy-chapman' to allow all defect species to redistribute.
- **initial_guess** (*list*) – Values for an initial guess for the defect mole fractions used in the error minimisation.

mole_fraction_error(*input_mole_fractions*, *target_mole_fractions*, *approximation*)
Calculates the sum of sqaures error between the output mole fractions calculated from the input mole fractions, compared to the target mole fractions.

Parameters

- **input_mole_fractions** (*list*) – Mole fractions for each of the species used in the iterative Poisson-Boltzmann solver.
- **target_mole_fractions** (*list*) – The value that the mole fractions should be in the bulk.

Returns Sum of squares error between output and target mole fractions.

Return type float

mole_fraction_output(*input_mole_fractions*, *approximation*)
Calculates the output mole fraction for a given input mole fraction when solving the Poisson-Boltzmann equation.

Parameters

- **input_mole_fractions** (*list*) – Mole fractions for each of the species used in the iterative Poisson-Boltzmann solver.
- **approximation** (*str*) – The defect mobility approximation. Either 'mott-schottky' to enforce only a single mobile defect, or 'gouy-chapman' to allow all defect species to redistribute.

Returns Mole fractions that are calculated from the iterative Poisson-Boltzmann solver.

Return type list

mole_fractions()
Calculate the mole fractions (probability of defects occupation) for each site on the subgrid for each species. The output is stored as a Calculation attribute. Calculation.mf (dict): A dictionary of the defect species mole fractions for each site on the subgrid for each site species. :param None:

solve(*approximation*)
Self-consistent solving of the Poisson-Boltzmann equation. Iterates until the convergence is less than the convergence limit. The outputs are stored as Calculation attributes. Calculation.phi (array): Electrostatic

potential on a one-dimensional grid. Calculation.rho (array): Charge density on a one-dimensional grid. Calculation.niter (int): Number of iterations performed to reach convergence.

Parameters **approximation** (*str*) – Approximation used for the defect behaviour. 'mott-schottky' - Some defects immobile / fixed to bulk mole fractions. 'gouy-chapman' - All defects mobile / able to redistribute.

solve_MS_approx_for_phi (*valence*)

Calculate the space-charge potential, ϕ_0, from the grain-boundary resistivity ratio, within the Mott-Schottky approx

Within the Mott-Schottky approximation the grain boundary resistivity is related to the space-charge potential (the electrostatic potential at the grain boundary core, compared to the bulk value) according to

$$r_{\mathrm{gb}} =$$

rac{ ho_{i,mathrm{gb}}}{ ho_{i,infty}} = rac{exp(z_iphi_0 / V_mathrm{th})}{2z_iphi_0/V_mathrm{th}}

where

$$V_{\mathrm{th}} =$$

rac{k_mathrm{B}T}{q}.

(See e.g. S. Kim, Phys. Chem. Chem. Phys. 18, 19787 (2016).)

This allows a Mott-Schottky space-charge potential, $\phi_{0,\mathrm{MS}}$, to be calculated using the LambertW function:

$$\phi_{0,\mathrm{MS}} = -\mathrm{LambertW}\,($$

rac{1}{2 r_mathrm{gb}} ight) rac{V_mathrm{th}}{z_i}.

The output is stored as a Calculation attribute. Calculation.ms_phi (float): $\phi_{0,\mathrm{MS}}$. The space charge potential calculated from Mott-Schottky model.

Args: valence(float): Charge of the mobile defect species.

Raises: ValueError: If the calculated resistivity ratio is less than 1.36, the LambertW function returns a complex, non-physical value.

pyscses.calculation.**calculate_activation_energies** (*ratios*, *temp*)

Solves the Arrhenius equation using the calculated resistivity ratio for a series of temperatures to calculate the activation energy for single defect segregation.

Uses a central difference approach so endpoints filled in with NaN to create an array with the same length as the args.

Parameters

- **ratios** (*list*) – Resistivity ratios calculated for a range of temperatures.
- **temp** (*list*) – Temperature (values used to calculate resistivity ratio values).

Returns The activation energy calculated for different temperatures.

Return type numpy.array

pyscses.calculation.**diff_central** (*x*, *y*)

Calculate the numerical derivative of x,y data using a central difference approximation.

Parameters

- **x** (*numpy.array*) – x values.
- **y** (*numpy.array*) – y values.

Returns numerical derivative of x,y

Return type numpy.array

pyscses.constants module

pyscses.defect_at_site module

class pyscses.defect_at_site.**Defect_at_Site**(*label*, *valence*, *mole_fraction*, *mobility*, *energy*, *site*, *fixed=False*)

The Defect_at_Site class contains the information about each defect at each site, its valence, mole fraction and the segregation energy for that defect at that site.

The methods in this class combine to give the correct statistics for site occupation in solid electrolytes, derived from the electrochemical potentials of a Fermi-Dirac like distribution. This term has been split into three functions for simplicity. The resulting equations take into account that the probability that a site is occupied by a defect can not exceed 1, and also accounts for competition of like charged defects. A Mott-Schottky approximation can be enforced. The defects can be fixed to their bulk mole fractions throughout the system, equivalent to assuming that the defects are immobile and not allowed to redistribute through the system.

label
string describing the defect species i.e. 'Vo' for an oxygen vacancy.

Type string

valence
The charge of the defect, in atomic units.

Type float

mole_fraction
The bulk mole fraction of the defect present in the system.

Type float

mobility
The bulk mobility of the defect species. Default = 0.0.

Type float

energy
The segregation energy for the defect when occupying the respective site.

Type float

site
The pyscses.Site object that corresponding to each defect at each x coordinate.

Type Site

fixed
set whether this defect species can redistribute to an equilibrium distriution. Default=False.

Type bool

boltzmann_one(*phi*, *temp*)
Boltzmann statistics calculation - part one

$$\exp\left(\frac{\Phi z + \Delta E}{k_B T}\right)$$

Parameters

- **phi** (*float*) – Electrostatic potential.
- **temp** (*float*) – Temperature of calculation.

Returns Boltzmann statistics

Return type float

boltzmann_three(*phi*, *temp*)
Boltzmann statistics calculation - part three

$$x\left(\exp\left(\frac{\Phi z + \Delta E}{K_B T}\right) - 1\right)$$

Parameters

- **phi** (*float*) – Electrostatic potential.
- **temp** (*float*) – Temperature of calculation.

Returns (Boltzmann statistics - 1) * mole fraction.

Return type float

boltzmann_two(*phi*, *temp*)
Boltzmann statistics calculation - part two

$$x\exp\left(\frac{\Phi z + \Delta E}{K_B T}\right)$$

Parameters

- **phi** (*float*) – Electrostatic potential.
- **temp** (*float*) – Temperature of calculation.

Returns Boltzmann statistics * mole fraction

Return type float

potential_energy(*phi*)
Potential energy for the defect at this site.

Parameters **phi** (*float*) – electrostatic potential at this site

Returns The electrochemical potential.

Return type float

pyscses.defect_species module

class pyscses.defect_species.**DefectSpecies**(*label*, *valence*, *mole_fraction*, *mobility=0.0*, *fixed=False*)
The DefectSpecies class includes the properties of each defect species present in the system.

Parameters

- **label** (*string*) – refers to what the defect is called i.e. 'Vo' for an oxygen vacancy.
- **valence** (*float*) – The charge of the defect, in atomic units.
- **mole_fraction** (*float*) – The bulk mole fraction of the defect present in the system.

- **fixed** (*bool*) – Whether the defect species can redistribute to an equilibrium distribution or whether the defect species is considered immobile and fixed to it's bulk mole fraction. Default=False.
- **mobility** (*float*) – Mobility of the defect species. Default = 0.0.

pyscses.grid module

class pyscses.grid.**Grid**(*x_coordinates*, *b*, *c*, *limits*, *limits_for_laplacian*, *set_of_sites*)

average_site_energies(*method='mean'*)
Returns the average segregation energy for all grid points based on a specified method

Parameters **method** (*str*) – The method in which the average segregation energies will be calculated. 'mean' - Returns the sum of all values at that site divided by the number of values at that site. 'min' - Returns the minimum segregation energy value for that site (appropriate for low temperature calculations).

Returns Average segregation energies on a 1D grid.

Return type average site energies (np.array)

charge(*phi*, *temp*)
Calculates the overall charge at each point on a grid.

Parameters

- **phi** (*np.array*) – Electrostatic potential on a 1D grid.
- **temp** (*float*) – Temperature of calculation.

Returns Overall charge at each point on a 1D grid.

Return type charge (np.array)

defect_valences()

Returns Valences for each defect from self.defects

Return type (np.array)

classmethod grid_from_set_of_sites(*set_of_sites*, *limits*, *limits_for_laplacian*, *b*, *c*)
Creates a grid from a given Set_of_Sites object.

Parameters

- **set_of_sites** (*object*) – Set_of_Sites object containing a set of all Site objects.
- **limits** (*list*) – distance between the midpoint of the endmost sites and the midpoint of the next site outside of the calculation region for the first and last sites respectively.
- **limits_for_laplacian** (*list*) – distance between the endmost sites and the next site outside of the calculation region for the first and last sites respectively.
- **b** (*float*) – b dimension for every grid-point.
- **c** (*float*) – c dimension for every grid-point.

Returns Grid object for the given set of sites.

Return type *Grid*

interpolated_energies()

Returns The average site energies linearly interpolated onto a regularly spaced grid

Return type energies (list)

rho(*phi*, *temp*)
Calculates charge density at each point on a grid, by dividing the overall charge at that grid point by the grid point volume.

Parameters

- **phi** (*np.array*) – Electrostatic potential on a 1D grid.
- **temp** (*float*) – Temperature of calculation.

Returns The charge density at each point on a grid.

Return type rho (np.array)

subgrid(*subset_species*)
Creates a subgrid for each species.

Parameters **subset_species** (*str*) – Site species to separate into subgrid.

Returns Grid object for subset of data.

Return type *Grid*

class pyscses.grid.**Grid_Point**(*x*, *volume*)
The Grid_Point class contains the information and calculations for each grid point individually

average_site_energy(*method='mean'*)
Returns the average segregation energy for all sites based on a specified method

Parameters **method** (*str*) – The method in which the average segregation energies will be calculated. 'mean' - Returns the sum of all values at that site divided by the number of values at that site. 'min' - Returns the minimum segregation energy value for that site (appropriate for low temperature calculations).

Returns Average segregation energies on a 1D grid.

Return type average site energies (np.array)

pyscses.grid.**avg**(*energies*, *method='mean'*)
Returns the average segregation energy for a site based on a specified method

Parameters

- **energies** (*np.array*) – Segregation energies on 1D grid.
- **method** (*str*) – The method in which the average segregation energies will be calculated. 'mean' - Returns the sum of all values at that site divided by the number of values at that site. 'min' - Returns the minimum segregation energy value for that site (appropriate for low temperature calculations).

Returns Average segregation energies on a 1D grid.

Return type average site energies (np.array)

pyscses.grid.**closest_index**(*myList*, *myNumber*)
Assumes myList is sorted. Returns index of closest value to myNumber. If two numbers are equally close, return the index of the smallest number.

Parameters

- **myList** (*list*) – List of numbers to compare against. This should be sorted.
- **myNumber** (*float*) – The number to compare against myList.

Returns Index of position of number in myList which is closest to myNumber.

Return type pos (int)

pyscses.grid.**delta_x_from_grid**(*coordinates*, *limits*)

Calculates the distance between the midpoints of each consecutive site. Inserts the calculated distance to the next grid point outside of the calculation region to the first and last position as the delta_x value for the endmost sites. :param coordinates: 1D grid of ordered numbers over a region. :type coordinates: np.array :param limits: Distance between the midpoint of the endmost sites and the midpoint of the repective site outside of the calculation region. :type limits: list

Returns Distance between the midpoints of each consecutive site.

Return type delta_x (np.array)

pyscses.grid.**energy_at_x**(*energy*, *coordinates*, *x*)

Assigns each site x coordinate a grid point and returns the segregation energy at the grid point clostest to the x coordinate.

Parameters

- **energy** (*np.array*) – Segregation energies on 1D grid.
- **coordinates** (*np.array*) – 1D grid of ordered numbers over a region.
- **x** (*float*) – Site x coordinate.

Returns The segregation energy at the x coordinate with position [index].

Return type energy[index] (float)

pyscses.grid.**index_of_grid_at_x**(*coordinates*, *x*)

Assigns each site x coordinate to a position on a regularly or irregularly spaced grid. Returns the index of the grid point clostest to the value x

Parameters

- **coordinates** (*np.array*) – 1D grid of ordered numbers over a region.
- **x** (*float*) – Site x coordinate

Returns Index of grid position closest to the site x coordinate.

Return type closest_index (int)

pyscses.grid.**phi_at_x**(*phi*, *coordinates*, *x*)

Assigns each site x coordinate a grid point and returns the electrostatic potential at the grid point clostest to the x coordinate.

Parameters

- **phi** (*np.array*) – electrostatic potential on 1D grid.
- **coordinates** (*np.array*) – 1D grid of ordered numbers over a region.
- **x** (*float*) – Site x coordinate.

Returns The electrostatic potential at the x coordinate with position [index].

Return type phi[index] (float)

pyscses.matrix_solver module

class pyscses.matrix_solver.**MatrixSolver**(*grid*, *dielectric*, *temp*, *boundary_conditions='dirichlet'*)

Contains the functions for the finite difference methods used to solve the Poisson-Boltzmann equation on a one-dimensional grid.

laplacian_new_fullmatrix()
Creates the full tridiagonal matrix used to solve the Poisson-Boltzmann equation using a finite element method. deltax is taken as the width of each grid point, from the center point between itself and the next grid point on either side. Using a finite difference approximation the diagonal becomes -2.0 / (deltax_1 * deltax_2), the upper diagonal becomes 2.0 / ((delta_x1 + delta_x2) * delta_x2) and the lower diagonal becomes 2.0 / ((delta_x1 + delta_x2) * delta_x1). If boundary_conditions are 'periodic', the corner elements of the matrix are filled.

Parameters **None** –

Returns Full tridiagonal matrix.

Return type A (matrix)

laplacian_sparse()
Creates a sparse matrix from a full matrix by taking only the tridiagonal values. :param None:

Returns Sparse tridiagonal matrix.

Return type A (matrix)

solve(*phi_old*)
Uses matrix inversion to solve the Poisson-Boltzmann equation through finite difference methods. The defect mole fractions are calculated from the elctrostatic potential, the charge density is calculated from the defect mole fractions and the elctrostatic potential is then updated using the updated charge density. If boundary_conditions are 'periodic', the charge density is minimised before the matrix inversion.

Parameters **phi_old** (*array*) – Electrostatic potential on a one-dimensional grid.

Returns Updated electrostatic potential on a one-dimensional grid. rho (array): Updated charge density on a one-dimensional grid.

Return type predicted_phi (array)

pyscses.set_of_sites module

class pyscses.set_of_sites.**Set_of_Sites**(*sites*)
The Set_of_Sites object groups together all of the Site objects into one object and contains functions for the calculations that provide properties of all of the sites together rather than individually.

calculate_defect_density(*grid*, *phi*, *temp*)
Calculates the defect density at each site.

Parameters

- **grid** (*object*) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **phi** (*array*) – electrostatic potential on a one-dimensional grid.
- **temp** (*float*) – Absolute temperature.

Returns defect density for each site using their grid points

Return type array

calculate_energies_on_grid(*grid*, *phi*)
Returns an array of energies at their points on a one-dimensional grid.

Parameters

- **grid** (*object*) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.

- **phi** (*array*) – electrostatic potential on a one-dimensional grid.

Returns energies at their grid points

Return type array

calculate_probabilities(*grid*, *phi*, *temp*)
Calculates the probability of a site being occupied by its corresponding defect.

Parameters

- **grid** (*object*) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates.
- **phi** (*array*) – electrostatic potential on a one-dimensional grid.
- **temp** (*float*) – Absolute temperature.

Returns probabilities of defects occupying each site using their grid points

Return type array

classmethod core_width_analysis(*input_data*, *limits*, *defect_species*, *site_charge*, *core*, *temperature*)
Calculated the width of the 'core' region. This is given as the region where the segregation energies in the system are within a region of positive to negative kT.

Parameters

- **input_data** (*file*) – A .txt file where each line includes information about a site.
- **limits** (*list*) – Minimum and maximum x coordinates defining the calculation region.
- **defect_species** (*object*) – Class object containing information about the defect species present in the system.
- **site_charge** (*bool*) – The site charge refers to the contribution to the overall charge of a site given by the original, non-defective species present at that site. True if the site charge contribution is to be included in the calculation, False if it is not to be included.
- **core** (*str*) – Core definition. Allowed keywords: 'single' = Single segregation energy used to define the core. 'multi-site' = Layered segregation energies used to define the core while the energies fall in the region of positive and negative kT. 'all' = All sites between a minimum and maximum x coordinate used in calculation.
- **temperature** (*float*) – Temperature that the calculation is being run at.

Returns Distance between the minimum and maximum x coordinates where the segregation energy is in the range of positive to negative kT.

Return type float

form_continuum_sites(*x_min*, *x_max*, *n_points*, *b*, *c*, *defect_species*, *limits_for_laplacian*, *site_labels*, *defect_labels*)
Creates a Set_of_Sites object for sites interpolated onto a regular grid, this is equivalent to assuming a continuum approximation.

Parameters

- **all_sites** (*object*) – Orginal Set_of_Sites object from full data.
- **x_min** (*float*) – Minimum x coordinate value defining the calculation region.
- **x_max** (*float*) – Maximum x coordinate value defining the calculation region.
- **n_points** (*int*) – Number of points that the data should be interpolated on to.

- **b** (*float*) – b dimension for every grid point.
- **c** (*float*) – c dimension for every grid point.
- **defect_species** (*object*) – Class object containing information about the defect species present in the system.
- **for laplacian** (*limits*) – distance between the endmost sites and the midpoint of the next site outside of the calculation region for the first and last sites respectively.
- **site_labels** (*list*) – List of strings for the different site species.
- **defect_labels** (*list*) – List of strings for the different defect species.

Returns Sites interpolated onto a regular grid.

Return type *Set_of_Sites*

get_coords(*label*)

Returns a list of the x coordinates for all the sites wich contain a particular defect

Parameters **label** (*str*) – Label identifying the required defect species.

Returns List of sites for a specific defect species.

Return type list

classmethod set_of_sites_from_input_data(*input_data*, *limits*, *defect_species*, *site_charge*, *core*, *temperature*, *offset=0.0*)

Takes the data from the input file and creates a Set_of_Sites object for those sites. The input data file is a .txt file where each line in the file corresponds to a site. The values in each line are formatted and separated into the corresponding properties before creating a Site object for each site.

Parameters

- **input_data** (*file*) – A .txt file where each line includes the information about a site.
- **limits** (*list*) – Minimum and maximum x coordinated defining the calculation region.
- **defect_species** (*object*) – Class object containing information about the defect species present in the system.
- **site_charge** (*bool*) – The site charge refers to the contribution to the overall charge of a site given by the original, non-defective species present at that site. True if the site charge contribution is to be included in the calculation, False if it is not to be included.
- **core** (*str*) – Core definition. 'single' = Single segregation energy used to define the core. 'multi-site' = Layered segregation energies used to define the core while the energies fall in the region of positive and negative kT. 'all' = All sites between a minimum and maximum x coordinate used in calculation.
- **temperature** (*float*) – Temperature that the calculation is being run at.

Returns *Set_of_Sites* object for the input data.

Return type *Set_of_Sites*

subgrid_calculate_defect_density(*sub_grid*, *full_grid*, *phi*, *temp*)

Calculates the defect density at each site for a given subset of sites.

Parameters

- **subgrid** (*object*) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates. For a given subset of sites.
- **full_grid** (*object*) – Grid object - contains properties of the grid including the x coordinates and the volumes. Used to access the x coordinates. For all sites.

- **phi** (*array*) – electrostatic potential on a one-dimensional grid.
- **temp** (*float*) – Absolute temperature.

Returns defect density for each site using their grid points

Return type array

subset (*label*)
Returns a list of all the sites which contain a particular defect

pyscses.set_up_calculation module

pyscses.set_up_calculation.**average_similar_sites** (*input_data*)
Corrects for rounding errors in the input data. Sites within 0.02nm are taken as the same site, therefore the x coordinate values are averaged to prevent the formation of separate sites.

Parameters **input_data** (*str*) – The input file.

Returns The input file, formatted with x coordinate values within 0.02nm averaged and set to the same value.

Return type str

pyscses.set_up_calculation.**calculate_grid_offsets** (*filename*, *x_min*, *x_max*, *system*)
Reads in the input data calculates the distance to the next site outside of the defined calculation region. Allows calculation of the delta_x and volume values for the endmost grid points.

Parameters

- **filename** (*string*) – Filename for importing the input data.
- **x_min** (*float*) – Minimum x coordinate value defining the calculation region.
- **x_max** (*float*) – Maximum x coordinate value defining the calculation region.
- **system** (*str*) – 'single' for single grain boundary systems, 'double' for systems where the input data has been mirrored to give two grain boundaries.

Returns distance between the midpoint of the endmost sites and the midpoint of the next site outside of the calculation region for the first and last sites respectively. Distance between the endmost sites and the next site outside of the calculation region for the first and last sites respectively.

Return type list, list

pyscses.set_up_calculation.**format_line** (*line*, *site_charge*, *offset=0.0*)
Each line in the input file is formatted into separate site properties.

Parameters

- **line** (*str*) – A line in the input file.
- **site_charge** (*bool*) – The site charge refers to the contribution to the overall charge of a site given by the original, non-defective species present at that site. True if the site charge contribution is to be included in the calculation, False if it is not to be included.

Returns line from the input file, split into individual values.

Return type list

pyscses.set_up_calculation.**load_site_data** (*filename*, *x_min*, *x_max*, *site_charge*, *offset=0.0*)
Reads in the input data and formats the input data if the x coordinate values are within the calculation region.

Parameters

- **filename** (*string*) – Filename for importing the input data.
- **x_min** (*float*) – Minimum x coordinate value defining the calculation region.
- **x_max** (*float*) – Maximum x coordinate value defining the calculation region.
- **site_charge** (*bool*) – True if site charges are to be included in the calculation, false if they are not to be included.

Returns formatted data for calculation.

Return type list

pyscses.set_up_calculation.**site_from_input_file**(*site*, *defect_species*, *site_charge*, *core*, *temperature*)

Takes the data from the input file and converts it into a site. The input data file is a .txt file where each line in the file corresponds to a site. The values in each line are formatted and separated into the corresponding properties before creating a Site object for each site.

Parameters

- **site** (*str*) – A line in the input file.
- **defect_species** (*object*) – Class object containing information about the defect species present in the system.
- **site_charge** (*bool*) – The site charge refers to the contribution to the overall charge of a site given by the original, non-defective species present at that site. True if the site charge contribution is to be included in the calculation, False if it is not to be included.

Returns Site

pyscses.site module

class pyscses.site.**Site**(*label*, *x*, *defect_species*, *defect_energies*, *scaling=None*, *valence=0*)

The site class contains all the information about a given site and the defect occupying that site. This class contains functions for the calculations which correspond to each individual site, rather than the system as a whole.

Parameters

- **label** (*str*) – refers to what the defect is called i.e. 'Vo' for an oxygen vacancy.
- **x** (*float*) – x coordinate of the site.
- **defect_energies** (*list*) – List of segregation energies for all defects present at the site.
- **defect_species** (*list*) – List of defect species for all defects present at the site.
- **defects** (*list*) – List of Defect_at_Site objects, containing the properties of all individual defects at the site.
- **scaling** (*float*) – A scaling factor that can be applied in the charge calculation.
- **valence** (*float*) – The charge of the defect present at the site (in atomic units).
- **defects** – List of Defect_Species objects for all defects present at the site.
- **sites** (*list*) – List containing all x coordinates and corresponding defect segregation energies.

average_local_energy(*method='mean'*)
Returns the average local segregation energy for each site based on a specified method

Parameters **method** (*str*) – The method in which the average segregation energies will be calculated. 'mean' - Returns the sum of all values at that site divided by the number of values at that site. 'min' - Returns the minimum segregation energy value for that site (appropriate for low temperature calculations).

Returns Average segregation energies on the site coordinates grid.

Return type numpy.array

charge(*phi*, *temp*)
Calculates the overall charge in Coulombs at each site.

Parameters

- **phi** (*float*) – Electrostatic potential at this site.
- **temp** (*float*) – Temperature of calculation.

Returns The charge on a 1D grid.

Return type np.array

charge_boltz(*phi*, *temp*)
Calculates the charge in Coulombs at each site using Boltzmann probabilities.

Parameters

- **phi** (*float*) – Electrostatic potential at this site
- **temp** (*float*) – Temperature of calculation.

Returns The charge on a 1D grid.

Return type np.array

defect_valences()
Returns an array of valences for each defect from self.defects

defect_with_label(*label*)
Returns a list of defects which correspond to the given label

Parameters **label** (*str*) – Label to identify defect species.

Returns List of Defect_at_Site objects for a specific defect species.

Return type list

energies()
Returns a list of the segregation energies for each defect from self.defects

probabilities(*phi*, *temp*)
Calculates the probability of each site being occupied. Derived from the chemical potential term for a Fermi-Dirac like distribution.

Parameters

- **phi** (*float*) – Electrostatic potential at this site.
- **temp** (*float*) – Temperature of calculation.

Returns Probabilities of site occupation on a 1D grid.

Return type list

probabilities_boltz(*phi*, *temp*)
Calculates the probability of each site being occupied by a given defect. Derived from the chemical potential including a Boltzmann distribution.

Parameters

- **phi** (*float*) – Electrostatic potential at this site.
- **temp** (*float*) – Temperature of calculation.

Returns Probabilities of site occupation on a 1D grid.

Return type list

sum_of_boltzmann_three(*phi*, *temp*)
Calculates the sum of the calculated boltzmann_three values for each defect at each site. i

$$\sum(x_i \exp(-\Phi_x z/kT) - 1)$$

Parameters

- **phi** (*float*) – Electrostatic potential at the site.
- **temp** (*float*) – Temperature of calculation in Kelvin.

Returns The sum of Boltzmann terms.

Return type float

Module contents

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index

G

I

L

M

P

R

S

V