gsl-ASW

0.1

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Atom Class Reference

```
#include <atom.h>
```

Public Member Functions

- int get_Z ()

  Get nuclear charge.
- void set_pos (gsl_vector &r)

  Set atom position (cartersian)
- void set_MT (double mt)

  Set muffin tin radius.
- void set_AS (double as)

  Set atomic sphere radius.
- gsl_vector get_pos ()

  Get atomic position (cartesian)
- double get_MT ()

  Get muffin tin radius.
- double get_AS ()

  Get atomic sphere radius.
- **Atom** (gsl_vector &r, Logarithmic_mesh &mesh)
- **Atom** (double mt, double as, double z, gsl_vector &r, Logarithmic_mesh &mesh)

2.1.1 Detailed Description

A class for representing atoms in the cell.
Contains:
Z - Nuclear charge
AS - Atomic sphere radius
MT - Muffin tin radius
pos - position, in cartesian coordinates
mesh - logarithmic mesh to use in intraatomic calculations

The documentation for this class was generated from the following files:

- atom.h
- atom.cpp

2.2 Crystal Class Reference

The documentation for this class was generated from the following files:

- crystal.h
- crystal.cpp

2.3 Effective_potential Class Reference

The documentation for this class was generated from the following file:

- eff_pot.h

2.4 Ewald_integral Class Reference

```
#include <ewald_int.h>
```

Public Member Functions

- void set_ewald_param (double eta)

  Set the Ewald parameter.
- void set_kappa (double kappa)

  Set energy parameter.
- std::vector< double > evaluate (lm l, Logarithmic_mesh &mesh)

  Calculate the Ewald integral for angular quantum number (l.l, l.m) and evaluate it on every point in the mesh.
- std::vector< double > evaluate_comp (lm l, Logarithmic_mesh &mesh)

  Calculate the complementary Ewald integral for angular quantum number (l.l, l.m) and evaluate it on every point in the mesh.

2.4.1 Detailed Description

A class for the integral representation of Hankel functions
Contains:
ewald_param - Parameter separating long range part from short range one
kappa - Energy parameter used (usually kappa^2 = -0.015)
h -

The documentation for this class was generated from the following files:

- ewald_int.h
- ewald_int.cpp

2.5 lm Struct Reference

Public Attributes

- int **l**
- int **m**

The documentation for this struct was generated from the following file:

- spherical_fun.h

2.6 Logarithmic_mesh Class Reference

```
#include <log_mesh.h>
```

Public Member Functions

- **Logarithmic_mesh** (double radius, unsigned int num_points)
- **Logarithmic_mesh** (double A, double radius, unsigned int num_points)

Public Attributes

- std::vector< double > **r**
- std::vector< double > **r2**
- std::vector< double > **drx**
- double **A**

2.6.1 Detailed Description

A class for representing logarithmic meshes
Contains:
B - Parameter determining shape of mesh, calculated from **num_points** and **A**
r - r-values contained in mesh
r2 - r^2-values contained in mesh
drx - Derivative dr/dx evaluated in mesh
A - Parameter controlling spacing between points in mesh

The documentation for this class was generated from the following files:

- log_mesh.h
- log_mesh.cpp

2.7 Numerov_solver Class Reference

```
#include <numerov_solver.h>
```

Public Member Functions

- void set_v_eff (double(∗new_v_eff)(double r))

    Set effective potential.
- void set_v_at (double(∗new_v_at)(double r))

    Set atomic potential.
- std::vector< double > solve_left (Logarithmic_mesh &mesh, int i_lr, std::vector< double > &init_cond, double E)

    Solve radial Schrödinger equation starting from the outer edge of the sphere ending at the inflection point.
- std::vector< double > solve_right (Logarithmic_mesh &mesh, int i_lr, std::vector< double > &init_cond, double E)

    Solve radial Schrödinger equation starting from the leftmost point ending at the inflection point.
- std::vector< double > solve (Logarithmic_mesh &mesh, std::vector< double > &l_init, std::vector< double > &r_init, double &en, int n_nodes)

2.7.1 Detailed Description

A class for solving the radial Schrödinger equation using Numerov's method
Find approximate energy values using variational method ensuring the solution has the correct number of nodes

2.7.2 Member Function Documentation

2.7.2.1 solve()

```
std::vector< double > Numerov_solver::solve (
            Logarithmic_mesh & mesh,
            std::vector< double > & l_init,
            std::vector< double > & r_init,
            double & en,
            int n_nodes )
```

Combine solutions from the left and the right and make sure they have the correct number of nodes and are continuous and smooth at the inflection point

The documentation for this class was generated from the following files:

- numerov_solver.h
- numerov_solver.cpp

2.8 Structure_constant Class Reference

```
#include <structure_const.h>
```

Public Member Functions

- **Structure_constant** (int l_low, int l_int, double kappa, lm l1, lm l2, gsl_vector r)
- **Structure_constant** (int l_low, int l_int, lm l1, lm l2, gsl_vector r)
- **Structure_constant** (int l_low, lm l1, lm l2, gsl_vector r)
- **Structure_constant** (lm l1, lm l2, gsl_vector r)

Public Attributes

- double **val**
- double **dk_val**

Friends

- std::ostream & operator<< (std::ostream &, const Structure_constant &)

  Functionality for printing structure constants.

2.8.1 Detailed Description

A class for representing structure constants
Contains:
l_int - Maximum orbital angular momentum to be included for Bessel expansions
l_low - Maximumm orbital angular momentum to be included for Hankel expansions
l1, **l2** - Orbital angular momenta to couple via the structure constant
kappa - Energy parameter used (usually kappa^2 = -0.015)
r - Position of atom
val - Value of the structure constant
dk_val - Value of energy derivative of the structure constant

The documentation for this class was generated from the following files:

- structure_const.h
- structure_const.cpp

Index