
CPPE Documentation

Release 0.0.0

Maximilian Scheurer

Jan 11, 2019

Table of Contents

1	C++ API (Doxygen)	3
2	Indices and tables	13

CPPE has been interfaced to the follow quantum chemical program packages:

- Psi4
- PYSCF
- Q-Chem


```
enum libcppe::BorderType
    Values:

    rem

    redist
    {"Electrostatic", {"Electronic", "Nuclear", "Multipoles"}}, {"Polarization", {"Electronic", "Nuclear", "Multi-
    poles"}}}]

std::vector<Potential> libcppe::get_polarizable_sites (std::vector<Potential> potentials)

Eigen::Vector3d libcppe::smat_vec(const Eigen::VectorXd &mat, const Eigen::Vector3d &vec,
                                double alpha)

Eigen::VectorXd libcppe::Tk_tensor(int k, const Eigen::Vector3d &Rij,
                                   std::vector<Eigen::MatrixXi> &Tk_coeffs)

Eigen::VectorXd libcppe::multipole_derivative(int k, int l, const Eigen::Vector3d
                                             &Rji, Eigen::VectorXd Mkj,
                                             std::vector<Eigen::MatrixXi> &Tk_coeffs)

int libcppe::xyz2idx (int x, int y, int z)

double libcppe::T(const Eigen::Vector3d &Rij, int x, int y, int z, std::vector<Eigen::MatrixXi> &Cijn)

std::vector<Eigen::MatrixXi> libcppe::Tk_coefficients (int max_order)

double libcppe::factorial (int n)

void libcppe::make_df (unsigned k, std::vector<double> &df)

int libcppe::trinom (int i, int j, int k)

std::vector<double> libcppe::symmetry_factors (unsigned k)

std::vector<double> libcppe::prefactors (unsigned k)

std::vector<double> libcppe::prefactors_nuclei (unsigned k)

int libcppe::multipole_components (int k)
```

```
bool libcppe::sortbysec (const std::pair<int, double> &a, const std::pair<int, double> &b)
```

```
struct Atom
    #include <molecule.hh>
```

Public Functions

```
Atom (int an)
```

```
Atom (int an, double x, double y, double z)
```

```
Eigen::Vector3d get_pos ()
```

Public Members

```
int atomic_number
```

```
int charge
```

```
double m_x
```

```
double m_y
```

```
double m_z
```

```
struct BorderOptions
    #include <pe_options.hh>
```

Public Members

```
BorderType border_type = {rem}
```

```
double rmin = 2.2
```

```
int nredist = 1
```

```
int redist_order = 1
```

```
bool redist_pol = false
```

```
class CPPE
    #include <libcppe.hh>
```

Public Functions

```
CPPE ()
```

```
~CPPE ()
```

```
std::vector<Potential> read_potfile (std::string potfile_name)
```


Private Members

bool **m_genlint_initialized**

bool **m_pe_initialized**

int **m_nbas**

int **m_nnbas**

int **m_natoms**

class CppeState

#include <cppe_state.hh>

Public Functions

CppeState ()

CppeState (*PeOptions* options, *Molecule* mol, std::ostream &output_stream = std::cout)

~CppeState ()

void **set_options** (*PeOptions* options)

void **set_molecule** (*Molecule* mol)

void **set_potentials** (std::vector<*Potential*> potentials)

std::vector<*Potential*> **get_potentials** ()

PeEnergy &**get_energies** ()

void **set_energies** (*PeEnergy* energy)

void **calculate_static_energies_and_fields** ()

std::vector<double> **get_induced_moments** () const

Eigen::VectorXd **get_induced_moments_vec** () const

void **update_induced_moments** (Eigen::VectorXd *elec_fields*, bool *elec_only* = false)

size_t **get_polarizable_site_number** ()

std::vector<double> **get_static_fields** ()

void **print_summary** ()

Private Members

Eigen::MatrixXd **m_es_operator**
PE electrostatics operator.

PeEnergy **m_pe_energy**
PE Energy Container.

Molecule **m_mol**
core region molecule

```
std::vector<Potential> m_potentials
    vector with all site potentials

size_t m_polarizable_sites
    number of polarizable sites

Eigen::VectorXd m_nuc_fields
    electric fields from nuclei

Eigen::VectorXd m_multipole_fields
    electric fields from multipole moments

Eigen::VectorXd m_induced_moments
    Vector with induced moments.

PeOptions m_options

std::ostream &m_output_stream = std::cout
    Output stream for printing.

bool m_make_guess = true

class InducedMoments
    #include <electric_fields.hh>
```

Public Functions

```
InducedMoments (std::vector<Potential> potentials, PeOptions options)

~InducedMoments ()

void compute (const Eigen::VectorXd &total_fields, Eigen::VectorXd &induced_moments, bool
    make_guess, std::ostream &output_stream = std::cout)

Eigen::VectorXd compute (Eigen::VectorXd &total_fields, bool make_guess)
    overloads the compute method for induced moments and returns a copy of the induced moments vector
```

Private Members

```
std::vector<Potential> m_potentials
    vector with all site potentials

std::vector<Potential> m_polsites
    vector with all potentials of polarizable sites

size_t m_n_polsites
    number of polarizable sites

PeOptions m_options

struct Molecule : public std::vector<Atom>
    #include <molecule.hh>
```

Public Functions

```
Eigen::Vector3d get_atom_position (int atom)

~Molecule ()
```

Molecule &operator= (const *Molecule*&)

```
class Multipole
    #include <multipole.hh>
```

Public Functions

```
Multipole (unsigned k)
~Multipole ()
void add_value (double val)
void remove_trace ()
std::vector<double> &get_values ()
Eigen::VectorXd get_values_vec ()
```

Public Members

```
unsigned m_k
```

Private Members

```
std::vector<double> m_values
```

```
class MultipoleExpansion
    #include <multipole_expansion.hh>
```

Public Functions

```
MultipoleExpansion (Molecule core, std::vector<Potential> potentials)
~MultipoleExpansion ()
double calculate_interaction_energy ()
```

Private Members

```
Molecule m_mol
    core region molecule
std::vector<Potential> m_potentials
    vector with all site potentials
```

```
class MultipoleFields
    #include <electric_fields.hh>
```

Public Functions

MultipoleFields (std::vector<*Potential*> *potentials*)

~MultipoleFields ()

Eigen::VectorXd **compute** (bool *damp* = false)

Private Members

std::vector<*Potential*> **m_potentials**

vector with all site potentials

std::vector<*Potential*> **m_polsites**

vector with all potentials of polarizable sites

size_t **m_n_polsites**

number of polarizable sites

class NuclearFields

#include <electric_fields.hh>

Public Functions

NuclearFields (*Molecule* *mol*, std::vector<*Potential*> *potentials*)

~NuclearFields ()

Eigen::VectorXd **compute** (bool *damp_core* = false)

Private Members

std::vector<*Potential*> **m_potentials**

vector with all site potentials

std::vector<*Potential*> **m_polsites**

vector with all potentials of polarizable sites

size_t **m_n_polsites**

number of polarizable sites

Molecule **m_mol**

core region molecule

struct PeEnergy

#include <pe_energies.hh> PE Energy Container

Public Functions

PeEnergy ()

double **get** (std::string *energy_string*)

returns energy contribution from given string

void **set** (std::string *energy_string*, double *energy*)
sets the energy titled *energy_string* to *energy*

Return void

Parameters

- *energy_string*: name of the energy contribution
- *energy*: value

double **get_total_energy** ()
returns the total PE energy

Private Members

std::vector<*PeEnergyContribution*> **m_energies**

struct PeEnergyContribution
#include <pe_energies.hh> PE Energy Contribution

Public Functions

PeEnergyContribution (std::string *cat*, std::string *name*, double *val*)

Public Members

std::string **m_category**
category of the energy, either “Electrostatic” or “Polarization”

std::string **m_name**
name of the energy, “Electronic”, “Nuclear”, or “Multipoles”

double **m_value**
energy value

struct PeOptions
#include <pe_options.hh>

Public Members

std::string **potfile** = {"potential.pot"}

int **print_level** = 1

bool **damp_induced** = false

bool **damp_multipoles** = false

bool **damp_core** = false

double **damp_coeff_ind** = 2.1304

double **damp_coeff_mult** = 2.1304

double **damp_coeff_core** = 2.1304

```
bool zero_pol = false
bool zero_mul = false
int zero_mul_order = 1
int induced_thresh = 8
bool do_diis = true
int diis_maxiter = 50
double diis_start_norm = 1.0
bool pe_border = false
BorderOptions border_options = {}
```

```
class Polarizability
#include <multipole.hh>
```

Public Functions

```
Polarizability()
~Polarizability()
void add_value(double val)
Eigen::VectorXd get_values_vec()
std::vector<double> &get_values()
```

Private Members

```
std::vector<double> m_values
```

```
class Potential
#include <multipole.hh>
```

Public Functions

```
Potential(double x, double y, double z, int idx)
~Potential()
void add_multipole(Multipole mul)
void add_polarizability(Polarizability pol)
void add_exclusion(int excl)
bool excludes_site(int other_site)
std::vector<int> &get_exclusions()
std::vector<Multipole> &get_multipoles()
std::vector<Polarizability> &get_polarizabilities()
```

```
bool is_polarizable ()
```

```
Eigen::Vector3d get_site_position ()
```

Public Members

```
double m_x
```

```
double m_y
```

```
double m_z
```

```
int index
```

Private Members

```
std::vector<Multipole> m_multipoles
```

```
std::vector<Polarizability> m_polarizabilities
```

```
std::vector<int> m_exclusions
```

```
class PotfileReader
```

```
#include <potfile_reader.hh>
```

Public Functions

```
PotfileReader (std::string potfile_name)
```

```
~PotfileReader ()
```

```
std::vector<Potential> read ()
```

Private Members

```
std::string m_potfile
```

```
class PotManipulator
```

```
#include <pot_manipulation.hh>
```

Public Functions

```
PotManipulator (std::vector<Potential> potentials, Molecule mol, std::ostream &output_stream =  
                std::cout)
```

```
~PotManipulator ()
```

```
std::vector<Potential> manipulate (PeOptions &pe_options)
```

Private Members

`std::vector<Potential> m_potentials`

`Molecule m_mol`

`std::ostream &m_output_stream`

struct Site

Public Members

`double x`

`double y`

`double z`

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

L

- libcppe::Atom (C++ class), 4
- libcppe::Atom::Atom (C++ function), 4
- libcppe::Atom::atomic_number (C++ member), 4
- libcppe::Atom::charge (C++ member), 4
- libcppe::Atom::get_pos (C++ function), 4
- libcppe::Atom::m_x (C++ member), 4
- libcppe::Atom::m_y (C++ member), 4
- libcppe::Atom::m_z (C++ member), 4
- libcppe::BorderOptions (C++ class), 4
- libcppe::BorderOptions::border_type (C++ member), 4
- libcppe::BorderOptions::nredist (C++ member), 4
- libcppe::BorderOptions::redist_order (C++ member), 4
- libcppe::BorderOptions::redist_pol (C++ member), 4
- libcppe::BorderOptions::rmin (C++ member), 4
- libcppe::BorderType (C++ type), 3
- libcppe::CPPE (C++ class), 4
- libcppe::CPPE::~~CPPE (C++ function), 4
- libcppe::CPPE::CPPE (C++ function), 4
- libcppe::CPPE::m_gen1int_initialized (C++ member), 5
- libcppe::CPPE::m_natoms (C++ member), 5
- libcppe::CPPE::m_nbas (C++ member), 5
- libcppe::CPPE::m_nnbas (C++ member), 5
- libcppe::CPPE::m_pe_initialized (C++ member), 5
- libcppe::CPPE::read_potfile (C++ function), 4
- libcppe::CppeState (C++ class), 5
- libcppe::CppeState::~~CppeState (C++ function), 5
- libcppe::CppeState::calculate_static_energies_and_fields (C++ function), 5
- libcppe::CppeState::CppeState (C++ function), 5
- libcppe::CppeState::get_energies (C++ function), 5
- libcppe::CppeState::get_induced_moments (C++ function), 5
- libcppe::CppeState::get_induced_moments_vec (C++ function), 5
- libcppe::CppeState::get_polarizable_site_number (C++ function), 5
- libcppe::CppeState::get_potentials (C++ function), 5
- libcppe::CppeState::get_static_fields (C++ function), 5
- libcppe::CppeState::m_es_operator (C++ member), 5
- libcppe::CppeState::m_induced_moments (C++ member), 6
- libcppe::CppeState::m_make_guess (C++ member), 6
- libcppe::CppeState::m_mol (C++ member), 5
- libcppe::CppeState::m_multipole_fields (C++ member), 6
- libcppe::CppeState::m_nuc_fields (C++ member), 6
- libcppe::CppeState::m_options (C++ member), 6
- libcppe::CppeState::m_output_stream (C++ member), 6
- libcppe::CppeState::m_pe_energy (C++ member), 5
- libcppe::CppeState::m_polarizable_sites (C++ member), 6
- libcppe::CppeState::m_potentials (C++ member), 5
- libcppe::CppeState::print_summary (C++ function), 5
- libcppe::CppeState::set_energies (C++ function), 5
- libcppe::CppeState::set_molecule (C++ function), 5
- libcppe::CppeState::set_options (C++ function), 5
- libcppe::CppeState::set_potentials (C++ function), 5
- libcppe::CppeState::update_induced_moments (C++ function), 5
- libcppe::ens (C++ member), 3
- libcppe::factorial (C++ function), 3
- libcppe::get_polarizable_sites (C++ function), 3
- libcppe::InducedMoments (C++ class), 6
- libcppe::InducedMoments::~~InducedMoments (C++ function), 6
- libcppe::InducedMoments::compute (C++ function), 6
- libcppe::InducedMoments::InducedMoments (C++ function), 6
- libcppe::InducedMoments::m_n_polsites (C++ member), 6
- libcppe::InducedMoments::m_options (C++ member), 6
- libcppe::InducedMoments::m_polsites (C++ member), 6
- libcppe::InducedMoments::m_potentials (C++ member), 6
- libcppe::make_df (C++ function), 3
- libcppe::Molecule (C++ class), 6
- libcppe::Molecule::~~Molecule (C++ function), 6
- libcppe::Molecule::get_atom_position (C++ function), 6

libcppe::Molecule::operator= (C++ function), 7
 libcppe::Multipole (C++ class), 7
 libcppe::Multipole::~Multipole (C++ function), 7
 libcppe::Multipole::add_value (C++ function), 7
 libcppe::Multipole::get_values (C++ function), 7
 libcppe::Multipole::get_values_vec (C++ function), 7
 libcppe::Multipole::m_k (C++ member), 7
 libcppe::Multipole::m_values (C++ member), 7
 libcppe::Multipole::Multipole (C++ function), 7
 libcppe::Multipole::remove_trace (C++ function), 7
 libcppe::multipole_components (C++ function), 3
 libcppe::multipole_derivative (C++ function), 3
 libcppe::MultipoleExpansion (C++ class), 7
 libcppe::MultipoleExpansion::~MultipoleExpansion
 (C++ function), 7
 libcppe::MultipoleExpansion::calculate_interaction_energy
 (C++ function), 7
 libcppe::MultipoleExpansion::m_mol (C++ member), 7
 libcppe::MultipoleExpansion::m_potentials (C++ mem-
 ber), 7
 libcppe::MultipoleExpansion::MultipoleExpansion (C++
 function), 7
 libcppe::MultipoleFields (C++ class), 7
 libcppe::MultipoleFields::~MultipoleFields (C++ func-
 tion), 8
 libcppe::MultipoleFields::compute (C++ function), 8
 libcppe::MultipoleFields::m_n_polsites (C++ member), 8
 libcppe::MultipoleFields::m_polsites (C++ member), 8
 libcppe::MultipoleFields::m_potentials (C++ member), 8
 libcppe::MultipoleFields::MultipoleFields (C++ func-
 tion), 8
 libcppe::NuclearFields (C++ class), 8
 libcppe::NuclearFields::~NuclearFields (C++ function), 8
 libcppe::NuclearFields::compute (C++ function), 8
 libcppe::NuclearFields::m_mol (C++ member), 8
 libcppe::NuclearFields::m_n_polsites (C++ member), 8
 libcppe::NuclearFields::m_polsites (C++ member), 8
 libcppe::NuclearFields::m_potentials (C++ member), 8
 libcppe::NuclearFields::NuclearFields (C++ function), 8
 libcppe::PeEnergy (C++ class), 8
 libcppe::PeEnergy::get (C++ function), 8
 libcppe::PeEnergy::get_total_energy (C++ function), 9
 libcppe::PeEnergy::m_energies (C++ member), 9
 libcppe::PeEnergy::PeEnergy (C++ function), 8
 libcppe::PeEnergy::set (C++ function), 8
 libcppe::PeEnergyContribution (C++ class), 9
 libcppe::PeEnergyContribution::m_category (C++ mem-
 ber), 9
 libcppe::PeEnergyContribution::m_name (C++ member),
 9
 libcppe::PeEnergyContribution::m_value (C++ member),
 9
 libcppe::PeEnergyContribution::PeEnergyContribution
 (C++ function), 9
 libcppe::PeOptions (C++ class), 9
 libcppe::PeOptions::border_options (C++ member), 10
 libcppe::PeOptions::damp_coeff_core (C++ member), 9
 libcppe::PeOptions::damp_coeff_ind (C++ member), 9
 libcppe::PeOptions::damp_coeff_mult (C++ member), 9
 libcppe::PeOptions::damp_core (C++ member), 9
 libcppe::PeOptions::damp_induced (C++ member), 9
 libcppe::PeOptions::damp_multipoles (C++ member), 9
 libcppe::PeOptions::diis_maxiter (C++ member), 10
 libcppe::PeOptions::diis_start_norm (C++ member), 10
 libcppe::PeOptions::do_diis (C++ member), 10
 libcppe::PeOptions::induced_thresh (C++ member), 10
 libcppe::PeOptions::pe_border (C++ member), 10
 libcppe::PeOptions::potfile (C++ member), 9
 libcppe::PeOptions::print_level (C++ member), 9
 libcppe::PeOptions::zero_mul (C++ member), 10
 libcppe::PeOptions::zero_mul_order (C++ member), 10
 libcppe::PeOptions::zero_pol (C++ member), 9
 libcppe::Polarizability (C++ class), 10
 libcppe::Polarizability::~Polarizability (C++ function),
 10
 libcppe::Polarizability::add_value (C++ function), 10
 libcppe::Polarizability::get_values (C++ function), 10
 libcppe::Polarizability::get_values_vec (C++ function),
 10
 libcppe::Polarizability::m_values (C++ member), 10
 libcppe::Polarizability::Polarizability (C++ function), 10
 libcppe::Potential (C++ class), 10
 libcppe::Potential::~Potential (C++ function), 10
 libcppe::Potential::add_exclusion (C++ function), 10
 libcppe::Potential::add_multipole (C++ function), 10
 libcppe::Potential::add_polarizability (C++ function), 10
 libcppe::Potential::excludes_site (C++ function), 10
 libcppe::Potential::get_exclusions (C++ function), 10
 libcppe::Potential::get_multipoles (C++ function), 10
 libcppe::Potential::get_polarizabilities (C++ function), 10
 libcppe::Potential::get_site_position (C++ function), 11
 libcppe::Potential::index (C++ member), 11
 libcppe::Potential::is_polarizable (C++ function), 11
 libcppe::Potential::m_exclusions (C++ member), 11
 libcppe::Potential::m_multipoles (C++ member), 11
 libcppe::Potential::m_polarizabilities (C++ member), 11
 libcppe::Potential::m_x (C++ member), 11
 libcppe::Potential::m_y (C++ member), 11
 libcppe::Potential::m_z (C++ member), 11
 libcppe::Potential::Potential (C++ function), 10
 libcppe::PotfileReader (C++ class), 11
 libcppe::PotfileReader::~PotfileReader (C++ function),
 11
 libcppe::PotfileReader::m_potfile (C++ member), 11
 libcppe::PotfileReader::PotfileReader (C++ function), 11
 libcppe::PotfileReader::read (C++ function), 11
 libcppe::PotManipulator (C++ class), 11

libcppe::PotManipulator::~~PotManipulator (C++ function), [11](#)
libcppe::PotManipulator::m_mol (C++ member), [12](#)
libcppe::PotManipulator::m_output_stream (C++ member), [12](#)
libcppe::PotManipulator::m_potentials (C++ member), [12](#)
libcppe::PotManipulator::manipulate (C++ function), [11](#)
libcppe::PotManipulator::PotManipulator (C++ function), [11](#)
libcppe::prefactors (C++ function), [3](#)
libcppe::prefactors_nuclei (C++ function), [3](#)
libcppe::redist (C++ enumerator), [3](#)
libcppe::rem (C++ enumerator), [3](#)
libcppe::Site (C++ class), [12](#)
libcppe::Site::x (C++ member), [12](#)
libcppe::Site::y (C++ member), [12](#)
libcppe::Site::z (C++ member), [12](#)
libcppe::smat_vec (C++ function), [3](#)
libcppe::sortbysec (C++ function), [3](#)
libcppe::symmetry_factors (C++ function), [3](#)
libcppe::T (C++ function), [3](#)
libcppe::Tk_coefficients (C++ function), [3](#)
libcppe::Tk_tensor (C++ function), [3](#)
libcppe::trinom (C++ function), [3](#)
libcppe::xyz2idx (C++ function), [3](#)