libecpint Documentation

Release 1.0.7

Robert A. Shaw

Feb 18, 2023

CONTENTS

CHAPTER

ONE

ABOUT

libecpint is an open-source library for the calculation of integrals over effective core potentials (ECPs) in quantum chemistry software. It was written predominantly by Robert A. Shaw (@robertshaw383), with some assistance from J. Grant Hill (@Dr_GHill). It is currently used in at least three major quantum chemistry packages: Psi4 (@PSI_code), Serenity (@QCSerenity), and entos (@EntosAI).

1.1 Introduction

1.1.1 Overview

libecpint is a C++ library for the efficient evaluation of integrals over ab initio effective core potentials, using a mixture of generated, recursive code and Gauss-Chebyshev quadrature. It is designed to be standalone and generic, but is currently in development and may not be completely stable. If you experience any problems please raise an issue here; contributions and suggestions are also welcome.

This assumes ECPs and basis sets of the form usually seen in electronic structure calculations, namely those expanded in terms of Gaussian functions. The angular momentum of function that can be treated is in theory arbitrary, but is limited by your choice of maximum when the library is built.

1.1.2 Citing libecpint

If you use this library in your program and find it helpful, that's great! Any feedback would be much appreciated. If you publish results using this library, please consider citing the following paper detailing the implementation:

R. A. Shaw, J. G. Hill, J. Chem. Phys. 147, 074108 (2017); doi: [10.1063/1.4986887](http://dx.doi.org/10.1063/1.4986887)

A full bibtex citation can be found in CITATION in the main directory.

Please also cite the ECPs and basis sets you use.

1.1.3 Requirements

For the library

- A modern C++ compiler, at least C++11 standard library is required. This has currently only been tested with GCC (6.3.0 and above, but will in theory work with any > 4.9) and clang (9.0.0 and above). Intel compilers have been known to cause issues.
- CMake/CTest build tools v. >= 3.12
- Python (2.7 or above, including 3 and higher)

For the docs

- Doxygen
- Sphinx
- Breathe
- Exhale

For radial code regeneration

- Python 3.6 or above
- numpy
- sympy

1.1.4 License

libecpint is available under an MIT License, allowing for free and open use, reproduction, and modification of the library, so long as the copyright and license notices are preserved. The authors hold no liability for, and give no warranty against, results of the use of this software.

1.1.5 Support

If you have any problems or would like to make suggestions for improvements, please raise an issue on the github repo. We will endeavour to get back to you as soon as possible, but as "we" is predominantly just "me" (Robert), it may take a while.

Help is always welcome, and if you wish to make contributions to the code yourself, please take a look at the library API docs and have a go. Send a pull request with any enhancements!

1.2 Installation

1.2.1 Obtaining libecpint

The latest stable release of libecpint can always be found at the Github Repo.

It can be downloaded directly from there, or you can clone it locally using git with the command

```
git clone https://github.com/robashaw/libecpint.git
```

If you are a developer looking to make changes to the code, please fork the repo into your own version, and make a pull request when you think your changes are production ready. We will not accept any attempts to push directly into master.

1.2.2 Building

To build the library, do the following in the top of the source tree (out-of-source build is required!):

```
mkdir build
cd build
cmake [options] ..
make [-jn]
```

The `-jn` flag tells make to use `n` threads while compiling (e.g. `-j4` would use four threads), and is highly recommended if your computer can cope, as the generated code files can all be compiled independently of one another.

CMake Options

The options after cmake above can be included using the syntax *-DOPTION=value*. The pertinent options are as follows:

Option	Default	Description
CMAKE_CXX_FLAGS	N/A	Flags to pass to the C++ compiler. These will depend on which compiler you're using, but in general we *strongly* recommend passing optimisation flags, specifically `-O2` or `-O3`.
CMAKE_INSTALL_PREFIX	/usr/local	The directory where the library will be installed. You must have permissions to edit this directory.
LIBECPINT_MAX_L	5	The maximum angular momentum (in the orbital and ECP basis) that the code will be able to handle. The higher this is, the longer the build stage will take, although not significantly so. NOTE: If you want derivatives of up to `L`, this must be `L+n` where `n` is the order of derivative (1 or 2).
LIBECPINT_MAX_UNROL	1	NOW REDUNDANT. The max. angular momentum the code is unrolled up to. Increasing this will make the build much slower but no longer gives any noticeable advantage. We *strongly* recommend leaving this at its default value.

Documentation

This documentation can be generated locally via CMake by running

```
make docs
```

This requires the following to be available:

- Doxygen
- Sphinx
- Breathe
- Exhale

Regenerating the radial code

The recursive radial integral code has been pre-generated, as the current setting has been calibrated to balance accuracy and efficiency. If you would like to experiment (warning: after reading the paper), go into the directory `src/generated/radial`. Edit the top line of `unrol_radial.py` to change `MAX_UNROL_AM`, the maximum angular momentum to be unrolled. Then do the following:

```
python3 unrol_radial.py
./generate.sh
```

This will generate the simplified recursive integrals and then piece together the `radial_gen.cpp` file and place it in the correct location. It should be very safe (but not very efficient) to decrease `MAX_UNROL_AM`, but be prepared for things to break if you increase it too much.

1.2.3 Testing

To run all the tests, in the build directory run

```
make test
```

This will give results that look as follows:

```
Running tests...
Test project [build-dir]
      Start  1: MathUtil
 1/16 Test  #1: MathUtil .........................   Passed    0.01 sec
      Start  2: MultiArray
 2/16 Test  #2: MultiArray .......................   Passed    0.00 sec
      Start  3: Bessel
 3/16 Test  #3: Bessel ...........................   Passed    0.02 sec
      Start  4: GaussianShell
 4/16 Test  #4: GaussianShell ....................   Passed    0.01 sec
      Start  5: GaussianECP
 5/16 Test  #5: GaussianECP ......................   Passed    0.02 sec
      Start  6: GaussQuad
 6/16 Test  #6: GaussQuad ........................   Passed    0.01 sec
      Start  7: Generator
 7/16 Test  #7: Generator ........................   Passed    0.00 sec
```

(continues on next page)

(continued from previous page)

```
      Start  8: IntTest1
 8/16 Test  #8: IntTest1 .........................   Passed    0.04 sec
      Start  9: IntTest2
 9/16 Test  #9: IntTest2 .........................   Passed    0.02 sec
      Start 10: DerivTest1
10/16 Test #10: DerivTest1 .......................   Passed    0.04 sec
      Start 11: DerivTest2
11/16 Test #11: DerivTest2 .......................   Passed    0.12 sec
      Start 12: HessTest1
12/16 Test #12: HessTest1 ........................   Passed    0.12 sec
      Start 13: HessTest2
13/16 Test #13: HessTest2 ........................   Passed    0.11 sec
      Start 14: APITest1
14/16 Test #14: APITest1 .........................   Passed    0.03 sec
      Start 15: APITest2
15/16 Test #15: APITest2 .........................   Passed    0.13 sec
      Start 16: Type1Test
16/16 Test #16: Type1Test ........................   Passed    0.03 sec
100% tests passed, 0 tests failed out of 16
```

If any of these tests fails, the reasons for the failure can then be found in `build/Testing/Temporary/LastTest.log`. This should help you troubleshoot and problems. If you have followed these instructions and the tests are still failing, please raise an issue on Github, giving details of your environment, any options you gave, and the relevant contents of LastTest.log.

Stress test

To see how efficient your build is, you can make an additional test as follows:

```
make StressTest
```

This will compute integrals and derivatives for increasing clusters of silver atoms, giving timings for each. At the end it will give estimated scaling exponents. Our latest build (clang 12.0.0, -O3 flag, single core on MacBook Pro 2017) gave these results:

```
N: 2
  Initialisation...   done. TIME TAKEN:        0.158172 seconds
       Integrals...   done. TIME TAKEN:        0.116014 seconds
      1st derivs...   done. TIME TAKEN:        0.321756 seconds
      2nd derivs...   done. TIME TAKEN:        0.785262 seconds
N: 4
  Initialisation...   done. TIME TAKEN:        0.138962 seconds
       Integrals...   done. TIME TAKEN:         1.23218 seconds
      1st derivs...   done. TIME TAKEN:         4.16765 seconds
N: 6
  Initialisation...   done. TIME TAKEN:        0.128387 seconds
       Integrals...   done. TIME TAKEN:         4.34048 seconds
      1st derivs...   done. TIME TAKEN:         15.7498 seconds
Scaling of integrals: N**3.31
Scaling of 1st derivs: N**3.56
```

1.2.4 Installation

To install the library and share directory, run

```
make install
```

which will create the following files/directories:

```
${CMAKE_INSTALL_PREFIX}/lib/libecpint.a
${CMAKE_INSTALL_PREFIX}/include/libecpint.hpp
${CMAKE_INSTALL_PREFIX}/include/libecpint
${CMAKE_INSTALL_PREFIX}/share/libecpint
```

1.3 Usage

There are two main ways to use the libecpint library. In the high-level API, you pass details of the system (basis set, coordinates, ECPs) to libecpint, and it handles the computation of all the integrals and/or derivatives automatically, returning arrays of the (Cartesian) integrals. In the low-level API, you control the calculation of the integrals yourself, calling the relevant routines as and when you need them. We envisage that for most purposes, the high-level API will be more appropriate, and is easier to use.

1.3.1 Important note about ECP definitions

ECP powers (`u_ns`) are assumed to be including the `r^2` from the spherical Jacobian, as is the convention followed with the Stuttgart-Dresden ECPs. In the code, we subtract two from this (see the ECP constructor) as integration is done in Cartesian coordinates. In general, the inputted `n` should only ever be 0, 1, or (most commonly) 2. If your ECP definitions are instead -2 or -1, or predominantly 0, that suggests you are following the other convention of not including the Jacobian, so you should add 2. Yes, this is annoying; if in doubt check the built-in ECP library for examples.

High Level API

Examples of using this API can be found in `tests/lib/api_test1` and `tests/lib/api_test2`.

1.3.2 The ECPIntegrator Object

The first step in using this API is to include the libecpint header and create an `ECPIntegrator` object:

```
include <libecpint.hpp>
ECPIntegrator factory;
```

This object will form the main interface to all of the subsequent routines, and is described in detail in the Library API.

1.3.3 Initialisation

There are three steps to initialising the ECPIntegrator before it can be used to calculate integrals. These are:

1) specifying the Gaussian basis set
2) specifying the ECP basis
3) calling the init routine

These steps are performed as follows:

```
factory.set_gaussian_basis(N_shells, g_coords, g_exps, g_coefs, g_ams, g_lengths);
factory.set_ecp_basis(N_ecps, u_coords, u_exps, u_coefs, u_ams, u_ns, u_lengths);
factory.init(deriv_order);
```

where N_shells and N_ecps are the numbers of shells in the Gaussian basis, and the number of ECP centers, respectively, while deriv_order is the maximum derivatives needed (0, 1, or 2). The rest of the parameters are:

- the Cartesian coordinates *in Bohr* (g_coords, u_coords);
- the exponents, coefficients, and angular momenta (and powers, u_ns, for the ECPs) comprising the basis sets
- the number of exponents per shell (g_lengths, u_lengths)

tests/lib/api_test1 shows how to specify these for HBr in the AVDZ/AVDZ-PP basis.

NOTE: The atom order given in set_gaussian_basis fixes the atom order for all the derivatives, as will be described later.

1.3.4 The ECP library

Alternatively, step 2 can be replaced by reading the ECPs from the built-in library provided with libecpint. This can be found in share/libecpint. To do this, you call:

```
factory.set_ecp_basis_from_library(N_ecps, u_coords, u_charges, u_names, share_dir);
```

The new parameters are:

- u_charges a list of atomic numbers for the ECPs, corresponding to the centers in u_coords;
- u_names the ECP names for each ECP, e.g. ecp10mdf;
- share_dir the absolute path to the share/libecpint directory, which must be passed by you.

The currently available ECPs in the library (more being added soon), and the atoms they are available for, are given below:

Name	Atoms
ECP10MDF	K – Kr (Z = 19 – 36)
ECP28MDF	Rb – Xe (Z = 37 – 54)
ECP46MDF	Cs, Ba (Z = 55, 56)
ECP60MDF	Hf – Rn (Z = 72 – 86), Ac – U (Z = 89 – 92)
ECP78MDF	Fr, Ra (Z = 87, 88)
LANL2DZ	Na – La (Z = 11 – 57), Hf – Bi (Z = 72 – 83), U – Pu (Z = 92 – 94)

** TO ADD AN ECP TO THE LIBRARY **

1) Put the ECP in MOLPRO format in `share/libecpint/raw` as `NAME.ecp`, where `NAME` is the name of the ECP; make sure that any exponents are with `E` (C-convention) not `D` (Fortran-convention).
2) Make the top line of `NAME.ecp` be the `NAME`.
3) In `share/libecpint` run `python3 parseecp.py NAME` (Python >=3.6 required, with lxml module). This will create `NAME.xml` in the `share/libecpint/xml` folder, and this ECP will now be available for use by libecpint. Please consider creating a pull request so that everyone can benefit from the addition!

Note that the `n` value for each ECP primitive should typically be 0, 1, or 2 (for the Stuttgart-Dresden ECPs, for example, it is *always* 2). Some input formats follow a convention of subtracting or adding 2 to this.

1.3.5 Computing integrals

Computing integrals over all shell pairs is then very simple:

```
factory.compute_integrals()
```

To retrieve these you then create a shared pointer to a vector:

```
std::shared_ptr<std::vector<double>> integrals = factory.get_integrals();
double I00 = (*ints)[0]; // example for accessing element (0, 0)
```

These are stored in row-order, and are in a *Cartesian Gaussian basis*. Typically these would be converted to a spherical harmonic Gaussian basis (we might add the ability to do this later). We follow canonical Cartesian order, so for a d-type function this would be `xx, xy, xz, yy, yz, zz`, and the order of the shells is the same as when you called `set_gaussian_basis`. The total number of Cartesian gaussians is stored in `factory.ncart`; you can access the ij-th integral as

```
double Iij = (*ints)[i*factory.ncart + j]
```

1.3.6 First derivatives

First derivatives are similarly calculated by calling

```
factory.compute_first_derivs()
```

Note that this will only work if `init` was called with `deriv_order > 0`. This will return an array of `3*factory.natoms` shared pointers to the integral derivatives with respect to each coordinate. The order is x, y, z, and the order of atoms matches that specified in `set_gaussian_basis`. For example, to get the array of integral derivatives with respect to the y-coordinate of the n-th atom, you would do:

```
std::vector<std::shared_ptr<std::vector<double>>> first_derivs = factory.get_first_
↪derivs();
I_dy_atom_n_00 = (*first_derivs[3*n+1])[0];
```

The order of the elements in each array is identical to that from `compute_integrals`.

1.3.7 Second derivatives

As for first derivatives, second derivatives are computed as

```
factory.compute_second_derivs()
```

and are provided as a vector of shared pointers to arrays. The order of these derivatives is somewhat more complicated though, and takes full advantage of symmetry. If the atoms are A, B, C, … as specified in `set_gaussian_basis`, then they are blocked as follows:

```
AA, AB, AC, ..., BB, BC, ..., CC, ...
```

Within each block the order is `xx, xy, xz, yy, yz, z` on the diagonal (e.g. AA, BB, CC, …) and `xx, xy, xz, yx, yy, yz, zx, zy, zz` on the off-diagonal (e.g. AB, AC, BC, …). There is a helper macro for this, `H_MACRO`, defined in `api.hpp`. So for example to get the derivative with respect to Ay (atom index 0) and Cx (atom index = 2) in a system with `N` atoms, you would do

```
int deriv_index = H_START(0, 2, N) + 3; // yx is index 3 for off-diagonal blocks
std::shared_ptr<std::vector<double>> h_Ay_Cx = factory.get_second_derivs()[deriv_index];
```

Hopefully this doesn't give you too much of a headache working out.

1.3.8 Updating coordinates

To update the coordinates for the basis and ECPs (for example, after a step in a geometry optimisation), simply pass the new coordinates *in the same order they were given when initialised*. This is done as:

```
factory.update_gaussian_basis_coords(N_shells, g_coords);
factory.update_ecp_basis_coords(N_ecps, u_coords);
```

You then call the compute routines when you need the new integrals and/or derivatives.

NOTE you will need to re-get the pointers using the get routines every time you recompute integrals/derivatives.

1.3.9 Settings

TODO detail optional settings that can be passed to ECPIntegrator

Low Level API

Examples of using this API can be found in `tests/lib/[name]_test[number]` where [name] is int (integrals), deriv (first derivatives), or hess (second derivatives), and [number] is 1 or 2.

1.3.10 The ECPIntegral Object

To be able to calculate integrals and derivatives at the shell-pair level, you need to create an `ECPIntegral` object instead. This is done as follows:

```
#include <libecpint/ecpint.hpp>
ECPIntegral ecpint(maxLB, maxLU, deriv_order);
```

where `maxLB` is the maximum angular momentum in the Gaussian basis, `maxLU` is the maximum angular momentum in the ECP basis, and `deriv_order` is the maximum order of derivative needed (defaults to 0).

1.3.11 Making shells and ECPs

The compute functions in this API require `ECP` and `GaussianShell` objects representing the ECP and Gaussian basis functions, respectively. These are populated for the ECP as follows:

```
#include <libecpint/ecp.hpp>
double C[3] = {Cx, Cy, Cz};
ECP newU(C);
newU.addPrimitive(n, l, x, c);
// addPrimitive for each primitive in ECP
```

where `C` is the coordinates of the center of the ECP (in Bohr), and `n, l, x, c` are the power, angular momentum, exponent, and coefficient of each primitive in that ECP. For the Gaussian basis instead:

```
#include <libecpint/gshell.hpp>
double A[3] = {Ax, Ay, Az};
GaussianShell shellA(A, l);
shellA.addPrim(x, c);
// addPrim for each primitive in this shell
```

where `A` is the coordinates of the center of the shell (in Bohr), and `l, x, c` are as above. You need to either create these objects for every shell and ECP as they are needed, or store them, so that they can be passed to the compute routines below.

1.3.12 Computing integrals

Computing integrals over a shell pair is then fairly simple. You call

```
#include <libecpint/mathutil.hpp>
TwoIndex<double> results;
ecpint.compute_shell_pair(newU, shellA, shellB, results);
```

This will place the integrals in a matrix, `results`, with dimensions `(ncartA, ncartB)`. The order in each is the canonical Cartesian order as described earlier for the high-level API. Elements of the results matrix can be accessed in two ways:

```
double Iij = results(i, j);
// or
double Iij = results.data[i*ncartB + j];
```

1.3.13 First derivatives

Similarly, first derivatives are calculated using

```
std::vector<TwoIndex<double>> results;
ecpint.compute_shell_pair_derivative(newU, shellA, shellB, results);
```

Now `results` is an array of 9 matrices (the matrices are ordered the same as for `compute_shell_pair` above). These are derivative matrices with respect to `Ax, Ay, Az, Bx, By, Bz, Cx, Cy, Cz`, where `A, B, C` are the centers of `shellA`, `shellB`, and `newU` respectively.

NOTE These derivatives are designed to be additive. Thus, if `A==B`, then the derivative for the Ax coordinate will be the sum `Ax+Bx`, etc.

1.3.14 Second derivatives

The second derivatives are calculated in the same simple manner, but their ordering is, as with the high-level API, much more complicated.

```
std::vector<TwoIndex<double>> results;
ecpint.compute_shell_pair_second_derivative(newU, shellA, shellB, results);
```

Now `results` contains **45** different derivative matrices. These are, in order:

```
AxAx, AxAy, AxAz, AyAy, AyAz, AzAz,
AxBx, AxBy, AxBz, AyBx, AyBy, AyBz, AzBx, AzBy, AzBz,
AxCx, AxCy, AxCz, AyCx, AyCy, AyCz, AzCx, AzCy, AzCz,
BxBx, BxBy, BxBz, ByBy, ByBz, BzBz,
BxCx, BxCy, BxCz, ByCx, ByCy, ByCz, BzCx, BzCy, BzCz,
CxCx, CxCy, CxCz, CyCy, CyCz, CzCz
```

where `A, B, C` are as described earlier. These are again additive but THERE IS NOW SYMMETRY TO CONSIDER. I *strongly* recommend that you look at the code in `api.cpp` for the construction of the full Hessian, to see how this symmetry has to be taken account of when any of the three centers are the same, as it is too complicated to describe here.

1.3.15 Settings

TODO detail optional settings that can be passed to ECPIntegral

1.4 Library API

1.4.1 Libecpint 1.0.2

page **`index`**

[ ]()

Libecpint is a C++ library for the efficient evaluation of integrals over ab initio effective core potentials, using a mixture of generated, recursive code and Gauss-Chebyshev quadrature. It is designed to be standalone and generic, and is now in its first stable release. If you experience any problems please raise an issue here; contributions and suggestions are also welcome.

New in first full release

- Analytical 1st and 2nd derivatives;
- Integration now >10x faster;
- New, high level API, with ECP library;
- Automated testing suite.

Patch 1

- Bug fix in screening of on-ECP type 2 integrals
- Improvements in CMake build steps, thanks to nabbelbabbel/moritzBens

Patch 2

- Fix for memory leaks in derivative routines
- Minor changes to CMake files

Dependencies

- A modern C++ compiler, at least C++11 standard library is required. This has been tested with:
  - gcc (v6.3.0 and above)
  - clang (v10.0.0 and above), you may need the CXX flag “-std=c++14”
  - icpc (v20.2.1), may also need the CXX flag “-std=c++14”
- CMake/CTest build tools (v3.12 and higher)
- Python (2.7 or above, including 3 and higher)

Additionally, if you wish to regenerate the radial code (see below), Python >=3.6 is required with numpy and sympy.

Documentation

Please refer to the main documentation here.

Acknowledging usage

If you use this library in your program and find it helpful, that’s great! Any feedback would be much appreciated. If you publish results using this library, please consider citing the following paper detailing the implementation:

R. A. Shaw, J. G. Hill, J. Chem. Phys. 147, 074108 (2017); doi: 10.1063/1.4986887

A full bibtex citation can be found in CITATION in the main directory.

Class Hierarchy

File Hierarchy

Full API

Namespaces

Namespace libecpint

Contents

Namespaces

- *Namespace libecpint::aux*
- *Namespace libecpint::qgen*

Classes

- *Struct ECP*
- *Struct ECPBasis*
- *Struct ECPIntegrator*
- *Template Struct FiveIndex*
- *Struct GaussianECP*
- *Struct GaussianShell*
- *Struct RadialIntegral::Parameters*
- *Template Struct SevenIndex*
- *Struct SumTerm*
- *Template Struct ThreeIndex*
- *Template Struct TwoIndex*
- *Class AngularIntegral*
- *Class BesselFunction*
- *Class ECPIntegral*

- *Class GCQuadrature*
- *Class RadialIntegral*
- *Class ShellPairData*

Enums

- *Enum GCTYPE*

Functions

- *Template Function libecpint::check_file*
- *Function libecpint::frobenius_norm*
- *Function libecpint::initFactorials*
- *Function libecpint::operator<<(std::ostream&, const SumTerm&)*
- *Template Function libecpint::operator<<(std::basic_ostream<Ch, Tr>&, std::tuple<Args…> const&)*
- *Function libecpint::pow_0*
- *Function libecpint::pow_1*
- *Function libecpint::pow_10*
- *Function libecpint::pow_11*
- *Function libecpint::pow_12*
- *Function libecpint::pow_13*
- *Function libecpint::pow_14*
- *Function libecpint::pow_15*
- *Function libecpint::pow_16*
- *Function libecpint::pow_17*
- *Function libecpint::pow_18*
- *Function libecpint::pow_19*
- *Function libecpint::pow_2*
- *Function libecpint::pow_20*
- *Function libecpint::pow_3*
- *Function libecpint::pow_4*
- *Function libecpint::pow_5*
- *Function libecpint::pow_6*
- *Function libecpint::pow_7*
- *Function libecpint::pow_8*
- *Function libecpint::pow_9*
- *Function libecpint::pow_m1*

- *Function libecpint::pow_m2*
- *Function libecpint::realSphericalHarmonics*
- *Function libecpint::shell_bound*

Typedefs

- *Typedef libecpint::Heptuple*
- *Typedef libecpint::Pair*
- *Typedef libecpint::Quintuple*
- *Typedef libecpint::Triple*

Variables

- *Variable libecpint::atom_names*
- *Variable libecpint::CAX1*
- *Variable libecpint::CAX2*
- *Variable libecpint::CAX3*
- *Variable libecpint::DFAC*
- *Variable libecpint::FAC*
- *Variable libecpint::FAST_POW*
- *Variable libecpint::GAMMA*
- *Variable libecpint::MAX_POW*
- *Variable libecpint::maxL*
- *Variable libecpint::maxN*
- *Variable libecpint::maxUnrol*
- *Variable libecpint::MIN_EXP*
- *Variable libecpint::ROOT_PI*
- *Variable libecpint::SINH_1*
- *Variable libecpint::SMALL*
- *Variable libecpint::TAYLOR_CUT*
- *Variable libecpint::TWO_C_TOLERANCE*

Namespace libecpint::aux

Contents

Classes

- *Template Struct gen_seq*
- *Template Struct gen_seq< 0, Is… >*
- *Template Struct seq*

Functions

- *Template Function libecpint::aux::print_tuple*

Namespace libecpint::qgen

Contents

Functions

- *Function libecpint::qgen::Q0_0_0*
- *Function libecpint::qgen::Q0_0_1*
- *Function libecpint::qgen::Q0_0_2*
- *Function libecpint::qgen::Q0_0_3*
- *Function libecpint::qgen::Q0_0_4*
- *Function libecpint::qgen::Q0_0_5*
- *Function libecpint::qgen::Q0_1_0*
- *Function libecpint::qgen::Q0_1_1*
- *Function libecpint::qgen::Q0_1_2*
- *Function libecpint::qgen::Q0_1_3*
- *Function libecpint::qgen::Q0_1_4*
- *Function libecpint::qgen::Q0_1_5*
- *Function libecpint::qgen::Q0_2_0*

- *Function libecpint::qgen::Q0_2_1*
- *Function libecpint::qgen::Q0_2_2*
- *Function libecpint::qgen::Q0_2_3*
- *Function libecpint::qgen::Q0_2_4*
- *Function libecpint::qgen::Q0_2_5*
- *Function libecpint::qgen::Q0_3_0*
- *Function libecpint::qgen::Q0_3_1*
- *Function libecpint::qgen::Q0_3_2*
- *Function libecpint::qgen::Q0_3_3*
- *Function libecpint::qgen::Q0_3_4*
- *Function libecpint::qgen::Q0_3_5*
- *Function libecpint::qgen::Q0_4_0*
- *Function libecpint::qgen::Q0_4_1*
- *Function libecpint::qgen::Q0_4_2*
- *Function libecpint::qgen::Q0_4_3*
- *Function libecpint::qgen::Q0_4_4*
- *Function libecpint::qgen::Q0_4_5*
- *Function libecpint::qgen::Q0_5_0*
- *Function libecpint::qgen::Q0_5_1*
- *Function libecpint::qgen::Q0_5_2*
- *Function libecpint::qgen::Q0_5_3*
- *Function libecpint::qgen::Q0_5_4*
- *Function libecpint::qgen::Q0_5_5*
- *Function libecpint::qgen::Q1_1_0*
- *Function libecpint::qgen::Q1_1_1*
- *Function libecpint::qgen::Q1_1_2*
- *Function libecpint::qgen::Q1_1_3*
- *Function libecpint::qgen::Q1_1_4*
- *Function libecpint::qgen::Q1_1_5*
- *Function libecpint::qgen::Q1_2_0*
- *Function libecpint::qgen::Q1_2_1*
- *Function libecpint::qgen::Q1_2_2*
- *Function libecpint::qgen::Q1_2_3*
- *Function libecpint::qgen::Q1_2_4*
- *Function libecpint::qgen::Q1_2_5*
- *Function libecpint::qgen::Q1_3_0*

- *Function libecpint::qgen::Q1_3_1*
- *Function libecpint::qgen::Q1_3_2*
- *Function libecpint::qgen::Q1_3_3*
- *Function libecpint::qgen::Q1_3_4*
- *Function libecpint::qgen::Q1_3_5*
- *Function libecpint::qgen::Q1_4_0*
- *Function libecpint::qgen::Q1_4_1*
- *Function libecpint::qgen::Q1_4_2*
- *Function libecpint::qgen::Q1_4_3*
- *Function libecpint::qgen::Q1_4_4*
- *Function libecpint::qgen::Q1_4_5*
- *Function libecpint::qgen::Q1_5_0*
- *Function libecpint::qgen::Q1_5_1*
- *Function libecpint::qgen::Q1_5_2*
- *Function libecpint::qgen::Q1_5_3*
- *Function libecpint::qgen::Q1_5_4*
- *Function libecpint::qgen::Q1_5_5*
- *Function libecpint::qgen::Q2_2_0*
- *Function libecpint::qgen::Q2_2_1*
- *Function libecpint::qgen::Q2_2_2*
- *Function libecpint::qgen::Q2_2_3*
- *Function libecpint::qgen::Q2_2_4*
- *Function libecpint::qgen::Q2_2_5*
- *Function libecpint::qgen::Q2_3_0*
- *Function libecpint::qgen::Q2_3_1*
- *Function libecpint::qgen::Q2_3_2*
- *Function libecpint::qgen::Q2_3_3*
- *Function libecpint::qgen::Q2_3_4*
- *Function libecpint::qgen::Q2_3_5*
- *Function libecpint::qgen::Q2_4_0*
- *Function libecpint::qgen::Q2_4_1*
- *Function libecpint::qgen::Q2_4_2*
- *Function libecpint::qgen::Q2_4_3*
- *Function libecpint::qgen::Q2_4_4*
- *Function libecpint::qgen::Q2_4_5*
- *Function libecpint::qgen::Q2_5_0*

- *Function libecpint::qgen::Q2_5_1*
- *Function libecpint::qgen::Q2_5_2*
- *Function libecpint::qgen::Q2_5_3*
- *Function libecpint::qgen::Q2_5_4*
- *Function libecpint::qgen::Q2_5_5*
- *Function libecpint::qgen::Q3_3_0*
- *Function libecpint::qgen::Q3_3_1*
- *Function libecpint::qgen::Q3_3_2*
- *Function libecpint::qgen::Q3_3_3*
- *Function libecpint::qgen::Q3_3_4*
- *Function libecpint::qgen::Q3_3_5*
- *Function libecpint::qgen::Q3_4_0*
- *Function libecpint::qgen::Q3_4_1*
- *Function libecpint::qgen::Q3_4_2*
- *Function libecpint::qgen::Q3_4_3*
- *Function libecpint::qgen::Q3_4_4*
- *Function libecpint::qgen::Q3_4_5*
- *Function libecpint::qgen::Q3_5_0*
- *Function libecpint::qgen::Q3_5_1*
- *Function libecpint::qgen::Q3_5_2*
- *Function libecpint::qgen::Q3_5_3*
- *Function libecpint::qgen::Q3_5_4*
- *Function libecpint::qgen::Q3_5_5*
- *Function libecpint::qgen::Q4_4_0*
- *Function libecpint::qgen::Q4_4_1*
- *Function libecpint::qgen::Q4_4_2*
- *Function libecpint::qgen::Q4_4_3*
- *Function libecpint::qgen::Q4_4_4*
- *Function libecpint::qgen::Q4_4_5*
- *Function libecpint::qgen::Q4_5_0*
- *Function libecpint::qgen::Q4_5_1*
- *Function libecpint::qgen::Q4_5_2*
- *Function libecpint::qgen::Q4_5_3*
- *Function libecpint::qgen::Q4_5_4*
- *Function libecpint::qgen::Q4_5_5*
- *Function libecpint::qgen::Q5_5_0*

- *Function libecpint::qgen::Q5_5_1*
- *Function libecpint::qgen::Q5_5_2*
- *Function libecpint::qgen::Q5_5_3*
- *Function libecpint::qgen::Q5_5_4*
- *Function libecpint::qgen::Q5_5_5*
- *Function libecpint::qgen::rolled_up*
- *Function libecpint::qgen::rolled_up_special*

Namespace makelist

Contents

Variables

- *Variable makelist::file*
- *Variable makelist::max_am*
- *Variable makelist::prefix*

Namespace std

STL namespace.

Namespace sympy

Namespace unrol_radial

Contents

Classes

- *Class Qijk*

Functions

- *Function unrol_radial::algebraic_unrol*
- *Function unrol_radial::collect*
- *Function unrol_radial::parse*
- *Function unrol_radial::unrol*

Variables

- *Variable unrol_radial::f*
- *Variable unrol_radial::file*
- *Variable unrol_radial::MAX_UNROL_AM*
- *Variable unrol_radial::q*

Classes and Structs

Template Struct gen_seq

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Struct Documentation

template<std::size_t **N**, std::size_t... **Is**>

struct **gen_seq**

Template Struct gen_seq< 0, Is... >

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Inheritance Relationships

Base Type

- `public libecpint::aux::seq< Is... >` (*Template Struct seq*)

Struct Documentation

template<std::size_t... **Is**>

struct **gen_seq**<0, *Is*...> : public libecpint::aux::*seq*<*Is*...>

Template Struct seq

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Struct Documentation

template<std::size_t...>

struct **seq**

Struct ECP

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_ecp.hpp

Struct Documentation

struct **ECP**

Stores the details of an *ECP* expanded in terms of Gaussians and spherical harmonics.

Unnamed Group

inline *GaussianECP* &**getGaussian**(int i)

Parameters **i** – - the index of *GaussianECP* required

Returns a reference to the ith *GaussianECP*

inline const *GaussianECP* &**getGaussian**(int i) const

Public Functions

ECP()

Constructs an empty *ECP* (N = 0, L=-1, center_ = {0, 0, 0})

ECP(const double *_center)

Constructs an *ECP* at the given position

Parameters **_center** – - xyz coordinates of the *ECP*

ECP(const *ECP* &other)

Copy constructor.

void **addPrimitive**(const int n, const int l, const double a, const double d, const bool needSort = true)

Adds a new *GaussianECP* to the *ECP*

Parameters

- **n** – - power of r
- **l** – - angular momentum
- **a** – - exponent
- **d** – - coefficient
- **needSort** – - true = the GaussianECPs are sorted (if done once at the end, speeds up evaluation)

inline const double ***center**() const

Returns the xyz coordinates of the *ECP*

void **setPos**(const double x, const double y, const double z)

void **sort**()

Sort primitives according to angular momentum.

inline int **getN**() const

Returns the number of primitives in *ECP*

bool **noType1**() const

Returns true if the highest angular momentum functions have zero coefficients (e.g. Stuttgart-Dresden ECPs)

double **evaluate**(const double r, const int l) const

Evaluates the *ECP* at a given distance for a given angular momentum shell.

Parameters

- **r** – - the radius at which to evaluate
- **l** – - the angular momentum shell to evaluate over

Returns the value of the l-th angular momentum shell of the *ECP* at radius r

inline int **getL**() const

Returns the maximum angular momentum in the *ECP*

Public Members

std::vector<*GaussianECP*> **gaussians**

All the primitives in the *ECP* expansion.

int **N**

Number of Gaussians.

int **L**

Maximum angular momentum.

int **atom_id**

Internal id of the atom the *ECP* is on.

double **min_exp**

minimum exponent in the *ECP*

double **min_exp_l**[LIBECPINT_MAX_L + 1]

minimum exponent in each l-shell

int **l_starts**[LIBECPINT_MAX_L + 2]

starting index of each l-shell

std::array<double, 3> **center_**

xyz coordinates of the atom on which the *ECP* is located

Struct ECPBasis

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_ecp.hpp

Struct Documentation

struct **ECPBasis**

A lightweight container for a basis set of *ECP* objects.

Unnamed Group

ECP &**getECP**(const int i)

Parameters **i** – - index of *ECP* required

Returns a reference to the ith *ECP* in basis

const *ECP* &**getECP**(const int i) const

Public Functions

ECPBasis()

Constructs an empty *ECPBasis* (N = maxL = 0)

void **addECP**(const *ECP* &U, const int atom)

Adds an *ECP* to basis.

Parameters

- **U** – - the *ECP* to be added
- **atom** – - the index of that atom on which U is located

int **getECPCore**(const int q) const

Parameters **q** – - an atomic number

Returns the number of electrons in core of *ECP* for the atom with atomic number q, if defined, otherwise zero

inline int **getAtom**(int i) const

Parameters **i** – - the index of *ECP* of interest

Returns the index of the atom on which the ith *ECP* is located

inline int **getMaxL**() const

Returns the maximum angular momentum *GaussianECP* in the entire *ECP* basis

inline int **getN**() const

Returns the number of ECPs in basis

void **addECP_from_file**(const int q, const std::array<double, 3> &coords, const std::string &filename)

Creates and adds an *ECP* object to the basis by reading from the *ECP* library

Parameters

- **q** – - the atomic number of the atom
- **coords** – - the [x, y, z] coordinates (in bohr) of the *ECP*
- **filename** – - path to the XML file containing the basis specification

Public Members

std::map<int, int> **core_electrons**

A map of atomic number to the number of electrons in core of *ECP* for that atom.

Struct ECPIntegrator

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_api.hpp

Struct Documentation

struct **ECPIntegrator**

API object that stores and handles all data for computing *ECP* integrals and their derivatives.

This is a higher level interface than directly using the *ECPIntegral* objects to calculate individual integrals. To use it, you follow these steps: 1) specify the gaussian basis 2) specify the ecp basis (either as a stream or from the library) 3) initialise the integrator, specifying derivative order 4) compute the desired quantities

The order of results follows the order of the atoms/basis functions as you specify them in steps 1 and 2. If you update the positions of your atoms at any point you must update them through this interface. Results are obtained as a pointer to a stream of data.

Public Functions

inline **ECPIntegrator**()

Default constructor, sets default values

void **set_gaussian_basis**(int nshells, const double *coords, const double *exponents, const double *coefs, const int *ams, const int *shell_lengths)

Constructs a basis set of *GaussianShell* objects from a stream of coordinates, exponents, coefficients, and angular momenta. Determines maxLB and ncart, and sets basis_is_set to true. The order of the atoms in coords determines the order of atoms in computed derivatives.

Parameters

- **nshells** – - the number of angular momentum shells in the basis
- **coords** – - a stream of cartesian coordinates (in bohr) in xyz order for each shell (size should be 3*nshells)
- **exponents** – - a stream of primitive exponents for each shell (size should be sum of shell_lengths)
- **coefs** – - a stream of coefficients corresponding to each exponent in exponents
- **ams** – - the angular momentum of each shell (size should be nshells)
- **shell_lengths** – - the number of primitives in each shell (size should be nshells)

void **set_ecp_basis**(int necps, const double *coords, const double *exponents, const double *coefs, const int *ams, const int *ns, const int *shell_lengths)

Constructs an *ECPBasis* with *ECP* objects for each *ECP* from streams of data. Determines maxLU. The order of the atoms doesn't matter, and ids will be matched to the order from set_gaussian_basis.

Parameters

- **necps** – - the number of ECPs
- **coords** – - a stream of cartesian coordinates (in bohr) in xyz order for each *ECP* (size should be 3*necps)
- **exponents** – - a stream of primitive exponents for each *ECP* (size should be sum of shell_lengths)
- **coefs** – - a stream of coefficients corresponding to each exponent in exponents
- **ams** – - the angular momentum of each primitive in exponents/coefs
- **ns** – - the order of r multiplying each primitive - we follow the convention where 2 is the default
- **shell_lengths** – - the number of primitives in each *ECP* (size should be necps)

void **set_ecp_basis_from_library**(int necps, const double *coords, const int *charges, const std::vector<std::string> &names, const std::string &share_dir)

Constructs an *ECPBasis* with *ECP* objects for each *ECP*, from the built-in *ECP* library. The order of the atoms doesn't matter, and ids will be matched to the order from set_gaussian_basis. It will search for the file "share_dir + / + name + .xml"

Parameters

- **necps** – - the number of ECPs
- **coords** – - a stream of cartesian coordinates (in bohr) in xyz order for each *ECP* (size should be 3*necps)

- **`charges`** – - the atomic numbers of each *ECP* atom (in same order as coords, size necps)
- **`names`** – - the name of each *ECP*, in same order as charges, e.g. “ecp10mdf” (size necps)
- **`share_dir`** – - the location of the share directory with the ecp library (typically “PATH/share/libecpint/xml”)

void **`update_gaussian_basis_coords`**(int nshells, const double *coords)

Updates the positions of the GaussianShells. The order of the coordinates must match that when originally specified in set_gaussian_basis.

Parameters

- **`nshells`** – - the number of angular momentum shells in the basis - must match nshells from set_gaussian_basis
- **`coords`** – - a stream of cartesian coordinates (in bohr) in xyz order for each shell (size should be 3*nshells)

void **`update_ecp_basis_coords`**(int necps, const double *coords)

Updates the positions of the ECPs The order of the coordinates must match that when originally specified in set_ecp_basis/set_ecp_basis_from_library

Parameters

- **`necps`** – - the number of ECPs
- **`coords`** – - a stream of cartesian coordinates (in bohr) in xyz order for each *ECP* (size should be 3*necps)

void **`init`**(int deriv_ = 0)

Initialises the *ECPIntegral* object, and determines the atom ids for each *GaussianShell* and *ECP*. This must be called AFTER the ECP/Gaussian bases are set, but BEFORE calling any of the compute functions.

Parameters **`deriv_`** – - the maximum derivative order to be computed; affects whether compute_first/second_derivs can be called; default 0

void **`compute_integrals`**()

Computes the *ECP* integrals across all shell pairs, returning the results into the integrals matrix. The order of the shells is canonical cartesian order, and matches the order in which the shells were specified in set_gaussian_basis.

void **`compute_first_derivs`**()

Computes the first derivative of the *ECP* integrals with respect to each atomic coordinate, placing the results in first_derivs. The atom order matches that specified in set_gaussian_basis, in xyz order.

void **`compute_second_derivs`**()

Computes the second derivative of the *ECP* integrals with respect to each pair of atomic coordinates,

but taking into account symmetry of second derivatives. The atom order matches that specified in set_gaussian_basis. See the docs for second_derivs for detailed description of the order.

inline std::shared_ptr<std::vector<double>> **`get_integrals`**()

Returns a shared pointer to the underlying data for integrals The packing is such that M(i, j) = i*ncart + j.

inline std::vector<std::shared_ptr<std::vector<double>>> **`get_first_derivs`**()

Returns a vector (size 3*natoms) of shared pointers to the data for first_derivs the packing is the same as for get_integrals, and the order is Ax,Ay,Az,Bx,By,Bz, etc.

inline std::vector<std::shared_ptr<std::vector<double>>> **get_second_derivs**()

> **Returns** a vector (size 3*natoms*(3*natoms+1)/2) of shared pointers to the data for second_derivs the packing is the same as for get_integrals, and the order is that specified in the docs for second_derivs

Public Members

std::vector<*GaussianShell*> **shells**

> a container of the Gaussian basis shells

ECPBasis **ecps**

> a container for the ECPs

std::shared_ptr<*ECPIntegral*> **ecpint**

> pointer to the *ECP* integral engine

int **maxLB**

> the maximum angular momentum in the gaussian basis, determined by set_gaussian_basis

int **deriv**

> maximum derivative order to be calculated (defaults to 0)

int **ncart**

> total number of cartesian gaussians in the gaussian basis, determined by set_gaussian_basis

int **natoms**

> total number of distinct atoms, determined during init

double **min_alpha**

> the minimum exponent in the gaussian basis

bool **ecp_is_set**

> true if the ecp basis has been set, false by default

bool **basis_is_set**

> true if the gaussian basis has been set, false by default

TwoIndex<double> **integrals**

> Container for the calculated *ECP* integrals, in canonical Cartesian order i.e. for L=2 : {x^2, xy, xz, y^2, yz, z^2}

std::vector<*TwoIndex*<double>> **first_derivs**

> Container for the *ECP* 1st derivatives, matrices are in same order as integrals, order in the vector is {Ax, Ay, Az, Bx, By, Bz, … } where atom order {A, B, C, …} is the same as provided when calling set_gaussian_basis. Total length is 3*natoms

std::vector<*TwoIndex*<double>> **second_derivs**

Container for the *ECP* 2nd derivatives, matrices are in same order as integrals, order in the vector is {AA, AB, AC, . . . , BB, BC, . . . , CC, . . . } where atom order {A, B, C, . . . } is the same as provided when calling set_gaussian_basis. The coordinate order within this is {xx, xy, xz, yy, yz, zz} for AA, BB, CC, etc. and {xx, xy, xz, yx, yy, yz, zx, zy, zz} for AB, AC, BC, etc. Total length is therefore: 3*natoms*(3*natoms+1)/2

Template Struct FiveIndex

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Struct Documentation

template<typename **T**>

struct **FiveIndex**

Templated skeleton five index array for convenience.

Public Functions

inline *T* &**operator()**(const int i, const int j, const int k, const int l, const int m)

inline *T* **operator()**(const int i, const int j, const int k, const int l, const int m) const

inline **FiveIndex**()

inline **FiveIndex**(const int dim1, const int dim2, const int dim3, const int dim4, const int dim5)

inline **FiveIndex**(const *FiveIndex*<*T*> &other)

Public Members

int **dims**[5]

std::vector<*T*> **data**

Struct GaussianECP

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_ecp.hpp

Struct Documentation

struct `GaussianECP`

Describes a Gaussian of angular momentum l of the form d r^n e^{-ax^2}.

Public Functions

`GaussianECP()`

Default constructor, sets n=l=d=0, a=1.

`GaussianECP(const int n, const int l, const double a, const double d)`

Constructs a new *GaussianECP*.

Parameters

- **n** – - power of r
- **l** – - angular momentum
- **a** – - exponent
- **d** – - coefficient

`GaussianECP(const GaussianECP &other)`

Copy constructor.

Public Members

int **n**

Power of r.

int **l**

Angular momentum.

double **a**

Exponent.

double **d**

Coefficient.

Struct GaussianShell

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_gshell.hpp

Struct Documentation

struct **`GaussianShell`**

Lightweight container for contracted shell of Gaussian basis functions.

Public Functions

`GaussianShell`(double *A, int l)

Constructs a *GaussianShell* with pointer to coords

Parameters

- **`A`** – - xyz coordinates of shell
- **`l`** – - angular momentum of shell

`GaussianShell`(const std::array<double, 3> &A, int l)

Constructs a *GaussianShell* with a local copy of coords

Parameters

- **`A`** – - xyz coordinates of shell
- **`l`** – - angular momentum of shell

inline **`GaussianShell`**(const *GaussianShell* &other)

Copy constructor for a *GaussianShell*

Parameters **`other`** – - reference to the *GaussianShell* to be copied

void **`addPrim`**(double exp, double c)

Adds a Gaussian primitive to the shell

Parameters

- **`exp`** – - the exponent
- **`c`** – - the contraction coefficient

inline int **`nprimitive`**() const

Returns the number of primitives

inline int **`ncartesian`**() const

Returns the number of cartesian basis functions in a shell with this angular momentum

inline double ***`center`**() const

Returns the xyz coordinates of the shell

inline double **`exp`**(int i) const

Parameters **`i`** – - the index of the primitive of interest

Returns the exponent of the ith primitive

inline double **`coef`**(int i) const

Parameters **`i`** – - the index of the primitive of interest

Returns the contraction coefficient of the ith primitive inline int **am**() const

Returns the angular momentum of the shell

inline *GaussianShell* **copy**() const

Returns a copy of this *GaussianShell*

Public Members

std::vector<double> **exps**

Exponents of primitives.

std::vector<double> **coeffs**

Coefficients in contraction of primitives.

double ***centerVec**

Pointer to xyz coordinates of basis function (pointer so as to update if geometry changes)

bool **local_ptr**

true if the centerVec is a reference to localCenter, false otherwise

double **localCenter**[3]

Local copy of coords if there is nothing else to point to.

double **min_exp**

the minimum exponent in the shell

int **l**

Angular momentum of shell.

int **atom_id**

internal id of the atom the shell is on

Struct RadialIntegral::Parameters

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_radial.hpp

Nested Relationships

This struct is a nested type of *Class RadialIntegral*.

Struct Documentation

struct **Parameters**

struct to store all parameters needed in both type 1 and 2 integrations

Public Members

TwoIndex<double> **p**

Matrices of parameters needed in both type 1 and 2 integrations.

TwoIndex<double> **P**

TwoIndex<double> **P2**

TwoIndex<double> **K**

Template Struct SevenIndex

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Struct Documentation

template<typename **T**>

struct **SevenIndex**

Templated skeleton seven index array for convenience.

Public Functions

inline *T* &**operator()**(const int i, const int j, const int k, const int l, const int m, const int n, const int p)

inline *T* **operator()**(const int i, const int j, const int k, const int l, const int m, const int n, const int p) const

inline **SevenIndex**()

inline **SevenIndex**(const int dim1, const int dim2, const int dim3, const int dim4, const int dim5, const int dim6, const int dim7)

inline **SevenIndex**(const *SevenIndex*<*T*> &other)

Public Members

int **dims**[7]

int **mults**[6]

std::vector<*T*> **data**

Struct SumTerm

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Struct Documentation

struct **SumTerm**

Stores details of terms in *ECP* integral expansion.

Public Functions

inline bool **operator<**(const *SumTerm* &s) const

orders *SumTerm*'s by mu, then by radial indices, then by angular integral value

Parameters `s` – - term to compare with

Returns true if this term is less than the given term

inline bool **operator<=**(const *SumTerm* &s) const

Returns true if this term is less than or equal to the given term

inline bool **operator==**(const *SumTerm* &s) const

Returns true if the mu and radial indices are equal

inline int **ca_index**() const

Returns the compressed index of CA

inline int **cb_index**() const

Returns the compressed index of CB

inline std::string **to_string**(bool full = true)

Converts term to string with compressed indices.

inline *Heptuple* **compare**(const *SumTerm* &s) const

Compares two *SumTerm* objects

Parameters `s` – - the *SumTerm* to compare with

Returns a tuple of equalities (0 = false, 1 = true) in the order {mu, radial, SA, SB, ang, CA, CB}

Public Members

Pair **SA**

(l, m) for spherical harmonic on shellA

Pair **SB**

(l, m) for spherical harmonic on shellB

Triple **radial**

(N, l1, l2) radial integral required

Quintuple **CA**

(0, cartesian index, x, y, z) for binomial coefficient on shellA

Quintuple **CB**

(0, cartesian index, x, y, z) for binomial coefficient on shellB

double **ang**

Value of product of angular integrals.

int **mu**

Value of mu, where the *ECP* ang. momentum is lam, and mu can range from -lam .. lam.

int **na**

Index of cartesian function on shellA, in alpha order.

int **nb**

Index of cartesian function on shellB, in alpha order.

Friends

friend std::ostream &**operator<<**(std::ostream &os, const *SumTerm* &s)

Prints out a *SumTerm* without compressing the indices - currently preferred.

Template Struct ThreeIndex

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Struct Documentation

template<typename **T**>

struct **ThreeIndex**

Templated skeleton three index array for convenience.

Public Functions

inline *T* &**operator()**(const int i, const int j, const int k)

inline *T* **operator()**(const int i, const int j, const int k) const

inline **ThreeIndex**()

inline **ThreeIndex**(const int dim1, const int dim2, const int dim3)

inline **ThreeIndex**(const *ThreeIndex*<*T*> &other)

inline void **fill**(const *T* value)

Public Members

int **dims**[3]

std::vector<*T*> **data**

Template Struct TwoIndex

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Struct Documentation

template<typename **T**>

struct **TwoIndex**

Templated skeleton two index array for convenience.

Public Functions

inline *T* &**operator()**(const int i, const int j)

inline *T* **operator()**(const int i, const int j) const

inline void **assign**(int dim1, int dim2, *T* value)

inline *TwoIndex*<*T*> **transpose**() const

inline void **add**(const *TwoIndex*<*T*> &other)

inline void **multiply**(*T* k)

inline **TwoIndex**()

inline **TwoIndex**(const int dim1, const int dim2)

inline **TwoIndex**(const int dim1, const int dim2, const *T* value)

inline **TwoIndex**(const *TwoIndex*<*T*> &other)

Public Members

int **dims**[2]

std::vector<*T*> **data**

Class AngularIntegral

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_angular.hpp

Class Documentation

class **AngularIntegral**

Calculates and stores the angular integrals needed for *ECP* integration.

This should not usually be created directly, it is instead owned by an *ECPIntegral* object, so that integrals can be performed over multiple *ECP* centers without duplicating work.

Public Functions

ThreeIndex<double> **uklm**(int lam, int mu) const

Calculates all possible USP to spherical transformation coefficients for a given angular momentum

Parameters

- **lam** – - the angular momentum
- **mu** – - the subshell

Returns *ThreeIndex* of the values U_lam,mu(k, l, m)

FiveIndex<double> **makeU**() const

Builds the USP to spherical transformation coefficients for use in calculating the type 1 and 2 integrals

Returns *FiveIndex* of the coefficients U(lam, lam+mu, k, l, m)

AngularIntegral()

Default constructor creates empty object.

AngularIntegral(int LB, int LE)

Specified constructor calls init with given arguments

Parameters

- **LB** – - the maximum angular momentum of the orbital basis
- **LE** – - the maximum angular momentum of the *ECP* basis

void **init**(int LB, int LE)

Initialises the object, must be called before anything else if default constructor was used.

Parameters

- **LB** – - the maximum angular momentum of the orbital basis
- **LE** – - the maximum angular momentum of the *ECP* basis

void **compute**()

Computes the type 1 and 2 angular integrals

void **clear**()

TODO: Clears the W and omega arrays.

double **getIntegral**(int k, int l, int m, int lam, int mu) const

Returns the type 1 angular integral W(k, l, m, lam, mu)

Parameters

- **k** – - x index
- **l** – - y index
- **m** – - z index
- **lam** – - angular momentum
- **mu** – - subshell

Returns value of type 1 angular integral

double **getIntegral**(int k, int l, int m, int lam, int mu, int rho, int sigma) const

Returns the type 2 angular integral Omega(k, l, m, lam, mu, rho, sigma)

Parameters

- **k** – - x index
- **l** – - y index
- **m** – - z index
- **lam** – - angular momentum of current *ECP* shell
- **mu** – - subshell of lam
- **rho** – - angular momentum of current basis shell
- **sigma** – - subshell of rho

Returns value of type 2 angular integral

inline int ***getOmegaMults**()

inline const int ***getOmegaMults**() const inline int ***getOmegaDims**()

inline const int ***getOmegaDims**() const

inline std::vector<double> &**getOmegaData**()

inline const std::vector<double> &**getOmegaData**() const

bool **isZero**(int k, int l, int m, int lam, int mu, double tolerance) const

is W(k, l, m, lam, mu) zero to within a given tolerance?

bool **isZero**(int k, int l, int m, int lam, int mu, int rho, int sigma, double tolerance) const

is Omega(k, l, m, lam, mu, rho, sigma) zero to within a given tolerance?

Class BesselFunction

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_bessel.hpp

Class Documentation

class **BesselFunction**

Computes a modified spherical Bessel function of the first kind.

Uses pretabulation to calculate the Bessel function up to a given maximum angular momentum. Values are interpolated using local Taylor series.

REFERENCES: R. Flores-Moreno et al., J. Comput. Chem. 27 (2006), 1009 L.E. McMurchie, E. Davidson, J. Comput. Phys. 44 (1981), 289

Public Functions

BesselFunction()

Default constructor. Creates a blank object.

BesselFunction(int lMax, int N, int order, double accuracy)

Specified constructor. Will call init with given arguments.

~BesselFunction()

Destructor, cleans up K and C.

void **init**(int lMax, int N, int order, double accuracy)

Initialises and pretabulates the *BesselFunction* up to the given angular momentum.

Parameters

- **lMax** – - the maximum angular momentum needed
- **N** – - the maximum number of points to be used in pretabulation, suggested 1600
- **order** – - the order at which the expansion is cut off, suggested 200
- **accuracy** – - the tolerance below which a value is considered converged

void **calculate**(double z, int maxL, std::vector<double> &values) const

Calculates the Bessel function values at a given point up to a given angular momentum

Parameters

- **z** – - point at which to evaluate
- **maxL** – - maximum angular momentum needed; must be <= lMax for object
- **values** – - reference to vector in which to put the values for l = 0 to maxL

double **calculate**(double z, int L) const

Calculates the Bessel function value at a given point for a single angular momentum

Parameters

- **z** – - point at which to evaluate
- **L** – - angular momentum needed; must be <= lMax for object

double **upper_bound**(double z, int L) const

Calculates an upper bound to the Bessel function value at a given point for a given angular momentum

Parameters

- **z** – - point at which to evaluate
- **L** – - angular momentum needed; must be <= lMax for object

Class ECPIntegral

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_ecpint.hpp

Class Documentation

class **ECPIntegral**

Calculates *ECP* integrals.

Given an *ECP* basis, and orbital bases, this will calculate the *ECP* integrals over all *ECP* centers.

REFERENCES: (Shaw2017) R. A. Shaw, J. G. Hill, J. Chem. Phys. 147 (2017), 074108 (Flores06) R. Flores-Moreno et al., J. Comput. Chem. 27 (2006), 1009 (MM81) L. E. McMurchie and E. R. Davidson, J. Comp. Phys. 44 (1981), 289 - 301

Public Functions

void **makeC**(*FiveIndex*<double> &C, int L, const double *A) const

Constructs the coefficients in the binomial expansion (see REF. Shaw2017)

Parameters

- **C** – - reference to a *FiveIndex* array to store the results in
- **L** – - maximum angular momentum to go up to in expansion
- **A** – - xyz coordinates for the center to calculate over

ECPIntegral(int maxLB, int maxLU, int deriv = 0)

Creates an *ECP* integrator, initialising the radial and angular parts for subsequent calculations.

Parameters

- **maxLB** – - the maximum angular momentum in the orbital basis
- **maxLU** – - the maximum angular momentum in the *ECP* basis
- **deriv** – - the maximum order of derivative to be calculated (TODO: derivs currently being implemented)

void **type1**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, const *ShellPairData* &data, const *FiveIndex*<double> &CA, const *FiveIndex*<double> &CB, const *RadialIntegral*::*Parameters* &parameters, *TwoIndex*<double> &values) const

Calculates the type 1 integrals for the given *ECP* center over the given shell pair, using quadrature

Parameters

- **U** – - reference to *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **data** – - wrapper for data about shell pair
- **CA** – - binomial expansion coefficients for shellA, made with makeC
- **CB** – - binomial expansion coefficients for shellB, made with makeC
- **parameters** – - pre-calculated parameters for the radial integral
- **values** – - array in which results are returned

void **type2**(int l, const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, const *ShellPairData* &data, const *FiveIndex*<double> &CA, const *FiveIndex*<double> &CB, const *RadialIntegral*::*Parameters* &parameters, *ThreeIndex*<double> &values) const

Calculates the type 2 integrals for the given *ECP* center over the given shell pair

Parameters

- **l** – - angular momentum shell of *ECP* to calculate over
- **U** – - reference to *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **data** – - wrapper for data about shell pair
- **CA** – - binomial expansion coefficients for shellA, made with makeC
- **CB** – - binomial expansion coefficients for shellB, made with makeC
- **parameters** – - pre-calculated parameters for the radial integral
- **values** – - array in which results are returned

void **estimate_type2**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, const *ShellPairData* &data, double *results) const void **compute_shell_pair**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, *TwoIndex*<double> &values, int shiftA = 0, int shiftB = 0) const

Computes the overall *ECP* integrals over the given *ECP* center and shell pair. This is the lower level API, where you want finer control over the calculation. Results are returned with rows corresponding to shellA and cols to shellB, with the Cartesian functions in alpha order e.g. {xxx, xxy, xxz, xyy, xyz, xzz, yyy, yyz, yzz, zzz} l = 3

Parameters

- **U** – - reference to the *ECP* to calculate the integral over
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **values** – - reference to *TwoIndex* array where the results will be stored

void **compute_shell_pair_derivative**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, std::array<*TwoIndex*<double>, 9> &results) const

Computes the overall *ECP* integral first derivatives over the given *ECP* center, C, and shell pair (A | B) The results are placed in order [Ax, Ay, Az, Bx, By, Bz, Cx, Cy, Cz] and are calculated so that each component can always be added to the relevant total derivative. E.g. if A = B, then the contribution to the total derivative for that coordinate on the x axis will be Ax + Bx. The order for each derivative matrices matches that specified in compute_shell_pair

Parameters

- **U** – - reference to the *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **results** – - reference to array of 9 *TwoIndex* arrays where the results will be stored

void **compute_shell_pair_second_derivative**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, std::array<*TwoIndex*<double>, 45> &results) const

Computes the overall *ECP* integral second derivatives over the given *ECP* center, C, and shell pair (A | B) The results are placed in order [AA, AB, AC, BB, BC, CC] with components [xx, xy, xz, yy, yz, zz] for AA, BB, and CC, and [xx, xy, xz, yx, yy, yz, zx, zy, zz] for AB, AC, and BC. As for the first derivatives, the components are calculated such that they can usually be added to the relevant total derivative. However, this is more complicated than for first derivatives, especially in the instance where A=B. It's recommended to look at the compute_second_derivatives interface in api.cpp for how to handle this. The order for each derivative matrices matches that specified in compute_shell_pair

Parameters

- **U** – - reference to the *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **results** – - reference to array of 45 *TwoIndex* arrays where the results will be stored

void **left_shell_derivative**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, std::array<*TwoIndex*<double>, 3> &results) const

Worker function to calculate the derivative of the integral <A | C | B> with respect to A. This is given as <d_q A(l_q) | C | B> = l_q*<A(l_q-1) | C | B> - 2*mu*<A(l_q + 1) | C | B> where l_q is the angular momentum component of A in the q coordinate, and mu is the exponent of A.

Parameters

- **U** – - reference to the *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **results** – - reference to array of 3 *TwoIndex* arrays for the [x, y, z] derivatives

void **left_shell_second_derivative**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, std::array<*TwoIndex*<double>, 6> &results) const

Worker function to calculate the second derivatives of the integral <A | C | B> with respect to AA. This is given as <d_p d_q A(l_p, l_q) | C | B> = l_p*l_q*<A(l_p-1, l_q-1) | C | B> - 2*mu*l_p*<A(l_p-1, l_q+1) | C | B>

- 2*mu*l_q*<A(l_p+1, l_q-1) | C | B> + 4*mu^2*<A(l_p+1, l_q+1) | C | B >

Parameters

- **U** – - reference to the *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **results** – - reference to array of 6 *TwoIndex* arrays for the [xx, xy, xz, yy, yz, zz] derivatives

void **mixed_second_derivative**(const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, std::array<*TwoIndex*<double>, 9> &results) const

Worker function to calculate the second derivatives of the integral <A | C | B> with respect to AB. This is given as <d_p A(l_p) | C | d_q B(l_q)> = l_p*l_q*<A(l_p-1) | C | B(l_q-1)> - 2*mu_B*l_p*<A(l_p-1) | C | B(l_q+1)>

- 2*mu_A*l_q*<A(l_p+1) | C | B(l_q-1)> + 4*mu_A*mu_B*<A(l_p+1) | C | B(l_q+1) >

Parameters

- **U** – - reference to the *ECP*
- **shellA** – - the first basis shell (rows in values)
- **shellB** – - the second basis shell (cols in values)
- **results** – - reference to array of 9 *TwoIndex* arrays for the [xx, xy, xz, yx, yy, yz, zx, zy, zz] derivatives

Public Members

int **skipped**

int **zero**

int **nonzero**

Class GCQuadrature

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_gaussquad.hpp

Class Documentation

class **GCQuadrature**

Performs adaptive Gauss-Chebyshev quadrature of the second kind for any given function.

Stores the weights and abscissae for the quadrature, and provides two different methods to integrate on [-1, 1] Also contains means to transform the region of integration to [0, infinity) and [rmin, rmax]

REFERENCES: (Perez92) J.M. Perez-Jorda et al., Comput. Phys. Comm. 70 (1992), 271-284 (Perez93) J.M. Perez-Jorda et al., Comput. Phys. Comm. 77 (1993), 46-56 (Krack98) M. Krack, A.M. Koster, J. Chem. Phys. 108 (1998), 3226 - 3234 (Flores06) R. Flores-Moreno et al., J. Comput. Chem. 27 (2006), 1009-1019

Public Functions

GCQuadrature()

Default constructor, creates empty object.

GCQuadrature(const *GCQuadrature* &other)

Copy constructor, carbon copies all members.

void **initGrid**(int points, *GCTYPE* t)

Intialises the integration grid to the given number of points, and integration type. ONEPOINT will choose N = 2^n - 1 closest to the given number of points, whilst TWOPOINT will choose N= 3*2^n - 1 in the same way.

Parameters

- **points** – - maximum number of quadrature points to be used
- **t** – - the algorithm to be used (ONEPOINT / TWOPOINT)

std::pair<double, bool> **integrate**(std::function<double(double, const double*, int)> &f, const double *params, double tolerance, int start, int end) const

Integrates the given function (over [-1, 1] by default) to within the given tolerance.

Parameters

- **f** – - the function to be integrated
- **params** – - array of parameters for the function to be integrated
- **tolerance** – - change below which convergenced is considered to be achieved
- **start** – - the index of the first point used in the integration
- **end** – - the index of the last point used in the integration

Returns the integral (first) and true if integration converged, false otherwise (second)

void **transformZeroInf**()

Transforms the region of integration to [0, inf) using the logarithmic transformation of Krack98 void **transformRMinMax**(double z, double p)

Transforms region of integration to [rmin, rmax] using the linear transformation from Flores06, assuming a Gaussian envelope. rmin/rmax are the distances from the centre of the envelope such that the integrand is effectively zero.

Parameters

- **z** – - the exponent of the Gaussian envelope
- **p** – - the centre of the Gaussian envelope

void **untransformRMinMax**(double z, double p)

inline int **getN**() const

Returns the maximum number of quadrature points

inline std::vector<double> &**getX**()

Returns a reference to the abscissae

inline const std::vector<double> &**getX**() const

Class RadialIntegral

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_radial.hpp

Nested Relationships

Nested Types

- *Struct RadialIntegral::Parameters*

Class Documentation

class **RadialIntegral**

Abstracts the calculation of radial integrals for *ECP* integration.

This should not be used directly, and is owned by *ECPIntegral*. It provides the interface to the adaptive quadrature algorithms used to calculate the type 1 and 2 radial integrals, and the unrolled recursive scheme for type 2 radial integrals.

Public Functions

RadialIntegral()

Default constructor creates an empty object.

void **init**(int maxL, double tol = 1e-15, int small = 256, int large = 1024)

Initialises the object, in turn intialising the quadrature grids and *BesselFunction*

Parameters

- **maxL** – - the maximum angular momentum of integral needed

- **tol** – - the tolerance for convergence of integrals (defaults to 1e-15)
- **small** – - the maximum number of quadrature points for the small integration grid (default 256, minimum recommended)
- **large** – - the maximum number of quadrature points for the large integration grid (default 1024, minimum recommended)

Parameters `buildParameters`(const *GaussianShell* &shellA, const *GaussianShell* &shellB, const *ShellPairData* &data) const

Given two GaussianShells, builds the parameters needed by both kind of integral.

Parameters

- **shellA** – - the first *GaussianShell*
- **shellB** – - the second *GaussianShell*
- **data** – - the data container for the shell pair

Returns the parameters needed in both type 1 and 2 integrations

void `type1`(int maxL, int N, int offset, const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, const *ShellPairData* &data, const *Parameters* &parameters, *TwoIndex*<double> &values) const

Calculates all type 1 radial integrals over two Gaussian shells up to the given maximum angular momentum.

Parameters

- **maxL** – - the maximum angular momentum
- **N** – - the power of r that the integrand is weighted by
- **offset** – - the starting angular momentum
- **U** – - the *ECP* to be integrated over
- **shellA** – - the first *GaussianShell*
- **shellB** – - the second *GaussianShell*
- **data** – - the data container for the shell pair
- **parameters** – - pre-calculated parameters for the radial integral
- **values** – - the matrix to return the integrals in

void `type2`(int lam, int l1start, int l1end, int l2start, int l2end, int N, const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, const *ShellPairData* &data, const *Parameters* &parameters, *TwoIndex*<double> &values) const

Calculates all type 2 radial integrals over two Gaussian shells for the given *ECP* angular momentum l using quadrature

Parameters

- **lam** – - the *ECP* shell angular momentum to be calculated over
- **l1start** – - the angular momentum to start on for the first shell
- **l1end** – - the angular momentum to stop at for the first shell
- **l2start** – - the angular momentum to start on for the second shell
- **l2end** – - the angular momentum to stop at for the second shell
- **N** – - the power of r that the integrand is weighted by

- **U** – - the *ECP* to be integrated over
- **shellA** – - the first *GaussianShell*
- **shellB** – - the second *GaussianShell*
- **data** – - the data container for the shell pair
- **parameters** – - pre-calculated parameters for the radial integral
- **values** – - the matrix to return the integrals in

void **type2**(const std::vector<*Triple*> &triples, int nbase, int lam, const *ECP* &U, const *GaussianShell* &shellA, const *GaussianShell* &shellB, double A, double B, *ThreeIndex*<double> &radials) const

Calculates all the requested type 2 radial integrals using predominantly a recursive algorithm. In the triples, l1 must be less than or equal to l2. Symmetry means that for l1 > l2, {N, l1, l2} can be calculated as {N, l2, l1} but with shellA and shellB (And therefore also A and B) swapped.

Parameters

- **triples** – - vector of triples of form {N, l1, l2} of all required radial integrals
- **nbase** – - the maximum number of base integrals that will be needed (so only have to compute once)
- **lam** – - the *ECP* shell angular momentum to be calculated over
- **U** – - the *ECP* to be integrated over
- **shellA** – - the first *GaussianShell*
- **shellB** – - the second *GaussianShell*
- **A** – - the magnitude of the distance of shellA from the *ECP*
- **B** – - the magnitude of the distance of shellB from the *ECP*
- **radials** – - the array to return the integrals in, indexed as (N, l1, l2)

double **estimate_type2**(int N, int l1, int l2, double n, double a, double b, double A, double B) const

Estimates the value of the requested type 2 radial integral for prescreening, as described in ref. Shaw2017. The modal point is estimated by ignoring the ratios of bessel function derivatives - this gives an overestimate so is okay for screening, but would not be good for approximating the integral itself.

Parameters

- **N** – - power of r in integrand
- **l1** – - angular momentum of first Bessel function
- **l2** – - angular momentum of second Bessel function
- **n** – - exponent of *ECP*
- **a** – - exponent of primitive in shellA
- **b** – - exponent of primitive in shellB
- **A** – - magnitude of distance of shellA from *ECP*
- **B** – - magnitude of distance of shellB from *ECP*

Returns estimated value (upper bound) of the type 2 integral

struct **Parameters**

struct to store all parameters needed in both type 1 and 2 integrations

Public Members

TwoIndex<double> **p**

Matrices of parameters needed in both type 1 and 2 integrations.

TwoIndex<double> **P**

TwoIndex<double> **P2**

TwoIndex<double> **K**

Class ShellPairData

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_gshell.hpp

Class Documentation

class **ShellPairData**

Stores the (shifted, for derivs) angular momenta, number of cartesians in a shell pair, and shifted centers.

Public Members

int **LA**

Angular momentum of shellA, shifted if deriv > 0.

int **LB**

Angular momentum of shellB, shifted if deriv > 0.

int **maxLBasis**

Maximum angular momentum in the orbital basis.

int **ncartA**

No. of cartesian BFs in shellA.

int **ncartB**

No. of cartesian BFs in shellB

double **A**[3]

xyz coords of shellA relative to *ECP*

double **B**[3]

xyz coords of shellB relative to *ECP*

double **A2**

Square of distance from *ECP* to shellA.

double **Am**

Magnitude of distance from *ECP* to shellA.

double **B2**

Square of distance from *ECP* to shellB.

double **Bm**

Magnitude of distance from *ECP* to shellB.

double **RAB2**

Square of distance between shells A and B.

double **RABm**

Magnitude of distance between shells A and B.

bool **A_on_ecp**

True if Am == 0.

bool **B_on_ecp**

True if Bm == 0.

Class Qijk

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Class Documentation

class Qijk

Public Functions

__init__(*self*, *Ival=0*, *Jval=0*, *Kval=0*)

print(*self*)

print_simple(*self*)

print_fgh(*self*)

write_code(*self*, *f*)

simplify(*self*)

`sort`(*self*)

`eliminate`(*self*)

Public Members

`i`

`j`

`k`

`size`

`start`

`end`

`subq`

`terms`

`bases`

`f`

`ga`

`gb`

`h`

Enums

Enum GCTYPE

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_gaussquad.hpp

Enum Documentation

enum `libecpint::GCTYPE`

Different choices of integration algorithm, see references.

Values:

enumerator **`ONEPOINT`**

Described in Perez92.

enumerator **`TWOPOINT`**

Described in Perez93.

Functions

Function generate_lists

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generate.cpp

Function Documentation

Warning: doxygenfunction: Cannot find function “generate_lists” in doxygen xml output for project “libecpint” from directory: ../../doxygen/xml/

Template Function libecpint::aux::print_tuple

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Function Documentation

template<class **Ch**, class **Tr**, class **Tuple**, std::size_t... **Is**>
void `libecpint::aux::`**`print_tuple`**(std::basic_ostream<*Ch*, *Tr*> &os, *Tuple* const &t, *seq*<*Is*...>)

Template Function libecpint::check_file

- Defined in file__Users_robertshaw_devfiles_libecpint_include_testutil.hpp

Function Documentation

template<typename **T**>
int libecpint::**check_file**(std::string filename, std::vector<*T*> &results, double thresh = 1e-5, double precision = 1e-10)

A helper function for tests that reads in a file of values and compares them to those provided by the test, returning 0 on success or 1 on failure.

Template Parameters **T** – - the type of the value; must be pipeable from a stringstream.

Parameters

- **filename** – - the file to read data from
- **results** – - reference to the vector of calculated results

Returns 0 if the results agree with the file within 0.00005%, 1 otherwise

Function libecpint::frobenius_norm

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double libecpint::**frobenius_norm**(const *TwoIndex*<double> &mat)

Parameters **mat** – - a reference to a *TwoIndex* array

Returns the Frobenius norm of mat

Function libecpint::initFactorials

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

void libecpint::**initFactorials**()

Initialises the global factorial and double factorial arrays

Function libecpint::operator<<(std::ostream&, const SumTerm&)

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Function Documentation

std::ostream &libecpint::**operator<<**(std::ostream &os, const *SumTerm* &s)

Template Function libecpint::operator<<(std::basic_ostream<Ch, Tr>&, std::tuple<Args. . .> const&)

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Function Documentation

template<class **Ch**, class **Tr**, class ...**Args**>
auto libecpint::**operator<<**(std::basic_ostream<*Ch*, *Tr*> &os, std::tuple<*Args*...> const &t) ->
std::basic_ostream<*Ch*, *Tr*>&

Function libecpint::pow_0

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double libecpint::**pow_0**(const double z)

Function libecpint::pow_1

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double libecpint::**pow_1**(const double z)

Function libecpint::pow_10

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double libecpint::**pow_10**(const double z)

Function libecpint::pow_11

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_11`(const double z)

Function libecpint::pow_12

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_12`(const double z)

Function libecpint::pow_13

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_13`(const double z)

Function libecpint::pow_14

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_14`(const double z)

Function libecpint::pow_15

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_15(const double z)`

Function libecpint::pow_16

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_16(const double z)`

Function libecpint::pow_17

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_17(const double z)`

Function libecpint::pow_18

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_18(const double z)`

Function libecpint::pow_19

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_19(const double z)`

Function libecpint::pow_2

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_2`(const double z)

Function libecpint::pow_20

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_20`(const double z)

Function libecpint::pow_3

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_3`(const double z)

Function libecpint::pow_4

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_4`(const double z)

Function libecpint::pow_5

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::`**`pow_5`**(const double z)

Function libecpint::pow_6

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::`**`pow_6`**(const double z)

Function libecpint::pow_7

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::`**`pow_7`**(const double z)

Function libecpint::pow_8

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::`**`pow_8`**(const double z)

Function libecpint::pow_9

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::`**`pow_9`**(const double z)

Function libecpint::pow_m1

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_m1`(const double z)

Function libecpint::pow_m2

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

double `libecpint::pow_m2`(const double z)

Function libecpint::qgen::Q0_0_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q0_0_0`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_0_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q0_0_1`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_0_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_0_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_0_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_0_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_0_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_0_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_0_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_0_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_1_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_1_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_1_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_1_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_1_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_1_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_1_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_1_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_1_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_1_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_1_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_1_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_2_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_2_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_2_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_2_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_2_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_2_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_2_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_2_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_2_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_2_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_2_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_2_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_3_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_3_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_3_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_3_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_3_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_3_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_3_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_3_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_3_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_3_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_3_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_3_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_4_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_4_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_4_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_4_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_4_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_4_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_4_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_4_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_4_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_4_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_4_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_4_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_5_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_5_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_5_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_5_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_5_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q0_5_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_5_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_5_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_5_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_5_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q0_5_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q0_5_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_1_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_1_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_1_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_1_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_1_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_1_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_1_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_1_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_1_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_1_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_1_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_1_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_2_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_2_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_2_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_2_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_2_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_2_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_2_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_2_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_2_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_2_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_2_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_2_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_3_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_3_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_3_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_3_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_3_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_3_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_3_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_3_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_3_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_3_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_3_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_3_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_4_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_4_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_4_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_4_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_4_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_4_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_4_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_4_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_4_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_4_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_4_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_4_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_5_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_5_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_5_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_5_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_5_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_5_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_5_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_5_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_5_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q1_5_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q1_5_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q1_5_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_2_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_2_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_2_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_2_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_2_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_2_2`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_2_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_2_3`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_2_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_2_4`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_2_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_2_5`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_3_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q2_3_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_3_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q2_3_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_3_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q2_3_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_3_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_3_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_3_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_3_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_3_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_3_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_4_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_4_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_4_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_4_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_4_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_4_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_4_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_4_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_4_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_4_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_4_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_4_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_5_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_5_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_5_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q2_5_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_5_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_5_2`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_5_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_5_3`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_5_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q2_5_4`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q2_5_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q2_5_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_3_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_3_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_3_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_3_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_3_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_3_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_3_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_3_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_3_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_3_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_3_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_3_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_4_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q3_4_0`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_4_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q3_4_1`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_4_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q3_4_2`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_4_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::Q3_4_3`(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_4_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_4_4`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_4_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_4_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_5_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_5_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_5_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q3_5_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_5_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q3_5_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_5_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q3_5_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_5_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q3_5_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q3_5_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q3_5_5`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_4_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q4_4_0`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_4_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q4_4_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_4_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_4_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_4_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_4_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_4_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_4_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_4_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_4_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_5_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_5_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_5_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_5_1**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_5_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_5_2**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_5_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_5_3**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_5_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_5_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q4_5_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q4_5_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q5_5_0

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q5_5_0**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q5_5_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q5_5_1`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q5_5_2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q5_5_2`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q5_5_3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void `libecpint::qgen::`**`Q5_5_3`**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q5_5_4

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q5_5_4**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::Q5_5_5

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_qgen.hpp

Function Documentation

void libecpint::qgen::**Q5_5_5**(const *ECP*&, const *GaussianShell*&, const *GaussianShell*&, const *FiveIndex*<double>&, const *FiveIndex*<double>&, const *TwoIndex*<double>&, const *TwoIndex*<double>&, double, double, const *RadialIntegral*&, const *AngularIntegral*&, const *RadialIntegral*::*Parameters*&, *ThreeIndex*<double>&)

Function libecpint::qgen::rolled_up

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_qgen.cpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “libecpint::qgen::rolled_up” with arguments (int, int, int, const ThreeIndex<double>&, const FiveIndex<double>&, const FiveIndex<double>&, const TwoIndex<double>&, const TwoIndex<double>&, const AngularIntegral&, ThreeIndex<double>&) in doxygen xml output for project “libecpint” from directory: ../../doxygen/xml/. Potential matches:

```
- void rolled_up(const int lam, const int LA, const int LB, const ThreeIndex<double> &
↪radials, const FiveIndex<double> &CA, const FiveIndex<double> &CB, const TwoIndex
↪<double> &SA, const TwoIndex<double> &SB, const AngularIntegral &angint, ThreeIndex
↪<double> &values)
```

Function libecpint::qgen::rolled_up_special

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_qgen.cpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “libecpint::qgen::rolled_up_special” with arguments (int, int, int, const ThreeIndex<double>&, const FiveIndex<double>&, const TwoIndex<double>&, const AngularIntegral&, ThreeIndex<double>&) in doxygen xml output for project “libecpint” from directory: ../../doxygen/xml/. Potential matches:

```
- void rolled_up_special(const int lam, const int LA, const int LB, const ThreeIndex
↪<double> &radials, const FiveIndex<double> &CB, const TwoIndex<double> &SB, const␣
↪AngularIntegral &angint, ThreeIndex<double> &values)
```

Function libecpint::realSphericalHarmonics

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “libecpint::realSphericalHarmonics” with arguments (int, double, double) in doxygen xml output for project “libecpint” from directory: ../../doxygen/xml/. Potential matches:

```
- TwoIndex<double> realSphericalHarmonics(const int lmax, const double x, const␣
↪double phi)
```

Function libecpint::shell_bound

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_api.cpp

Function Documentation

double `libecpint::`**`shell_bound`**(const int la, const double alpha, const double A2, const double eta)

Function main

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generate.cpp

Function Documentation

> **Warning:** doxygenfunction: Cannot find function "main" in doxygen xml output for project "libecpint" from directory: ../../doxygen/xml/

Function unrol_radial::algebraic_unrol

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Function Documentation

`unrol_radial.`**`algebraic_unrol`**`(`*i*, *j*, *k*`)`

Function unrol_radial::collect

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Function Documentation

`unrol_radial.`**`collect`**`(`*q*, *Q*, *term*`)`

Function unrol_radial::parse

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Function Documentation

`unrol_radial.`**`parse`**`(`*term*`)`

Function unrol_radial::unrol

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Function Documentation

`unrol_radial.`**`unrol`**`(`*q*`)`

Variables

Variable libecpint::atom_names

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_ecp.hpp

Variable Documentation

const std::string `libecpint::`**`atom_names`**[109] = {"h", "he", "li", "be", "b", "c", "n", "o", "f", "ne", "na", "mg", "al", "si", "p", "s", "cl", "ar", "k", "ca", "sc", "ti", "v", "cr", "mn", "fe", "co", "ni", "cu", "zn", "ga", "ge", "as", "se", "br", "kr", "rb", "sr", "y", "zr", "nb", "mo", "tc", "ru", "rh", "pd", "ag", "cd", "in", "sn", "sb", "te", "i", "xe", "cs", "ba", "la", "ce", "pr", "nd", "pm", "sm", "eu", "gd", "tb", "dy", "ho", "er", "tm", "yb", "lu", "hf", "ta", "w", "re", "os", "ir", "pt", "au", "hg", "tl", "pb", "bi", "po", "at", "rn", "fr", "ra", "ac", "th", "pa", "u", "np", "pu", "am", "cm", "bk", "cf", "es", "fm", "md", "no", "lr", "rf", "db", "sg", "bh", "hs", "mt"}

names of each atom in order of atomic number

Variable libecpint::CAX1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Variable Documentation

const int `libecpint::`**`CAX1`** = *maxN* + 1

Variable libecpint::CAX2

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Variable Documentation

const int `libecpint::`**`CAX2`** = CAX1 * (*maxL* + 1)

Variable libecpint::CAX3

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Variable Documentation

const int libecpint::**CAX3** = CAX2 * (*maxL* + 1)

Variable libecpint::DFAC

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Variable Documentation

double libecpint::**DFAC**[MAX_DFAC]

Array of double factorials.

Variable libecpint::FAC

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Variable Documentation

double libecpint::**FAC**[MAX_FAC]

Array of factorials.

Variable libecpint::FAST_POW

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Variable Documentation

static double (*libecpint::**FAST_POW**[23])(double)

Array of function pointers to hand-coded x**n routines.

Variable libecpint::GAMMA

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Variable Documentation

const double `libecpint::`**`GAMMA`**[30]

Gamma function tabulation, where GAMMA[i] = Gamma((i+1)/2) e.g. GAMMA[0] = Gamma(1/2) = sqrt(Pi), GAMMA[1] = 0! = 1, GAMMA[2] = Gamma(3/2) = sqrt(Pi) / 2, etc.

Variable libecpint::MAX_POW

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Variable Documentation

constexpr int `libecpint::`**`MAX_POW`** = 20

Variable libecpint::maxL

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Variable Documentation

const int `libecpint::`**`maxL`** = LIBECPINT_MAX_L

Variable libecpint::maxN

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Variable Documentation

const int `libecpint::`**`maxN`** = 15

Variable libecpint::maxUnrol

- Defined in file__Users_robertshaw_devfiles_libecpint_include_generate.hpp

Variable Documentation

const int `libecpint::`**`maxUnrol`** = LIBECPINT_MAX_UNROL

Variable libecpint::MIN_EXP

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_radial.hpp

Variable Documentation

constexpr double `libecpint::`**`MIN_EXP`** = 0.002

Variable libecpint::ROOT_PI

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Variable Documentation

constexpr double `libecpint::`**`ROOT_PI`** = 1.772453850905516

square root of PI

Variable libecpint::SINH_1

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Variable Documentation

constexpr double `libecpint::`**`SINH_1`** = 1.1752011936

Variable libecpint::SMALL

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_bessel.hpp

Variable Documentation

constexpr double `libecpint::SMALL` = 1.0E-7

Numerical tolerance of z below which K_n(z) = { 1 (n=0), 0 otherwise }.

Variable libecpint::TAYLOR_CUT

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_bessel.hpp

Variable Documentation

constexpr int `libecpint::TAYLOR_CUT` = 5

Order of local Taylor series to be used in Bessel function expansion.

Variable libecpint::TWO_C_TOLERANCE

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_api.hpp

Variable Documentation

constexpr double `libecpint::TWO_C_TOLERANCE` = 1E-12

Variable makelist::file

- Defined in file__Users_robertshaw_devfiles_libecpint_src_makelist.py

Variable Documentation

```
makelist.file = open(prefix + "/qlist.txt", "w")
```

Variable makelist::max_am

- Defined in file__Users_robertshaw_devfiles_libecpint_src_makelist.py

Variable Documentation

```
makelist.max_am = int(sys.argv[1])
```

Variable makelist::prefix

- Defined in file__Users_robertshaw_devfiles_libecpint_src_makelist.py

Variable Documentation

```
makelist.prefix = str(sys.argv[2])
```

Variable unrol_radial::f

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Variable Documentation

```
unrol_radial.f = open('radial_gen.part2', 'w')
```

Variable unrol_radial::file

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Variable Documentation

```
unrol_radial.file
```

Variable unrol_radial::MAX_UNROL_AM

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Variable Documentation

```
unrol_radial.MAX_UNROL_AM = 4
```

Variable unrol_radial::q

- Defined in file__Users_robertshaw_devfiles_libecpint_src_generated_radial_unrol_radial.py

Variable Documentation

```
unrol_radial.q = algebraic_unrol(i, j, k)
```

Defines

Define FAC_INIT

- Defined in file__Users_robertshaw_devfiles_libecpint_src_lib_mathutil.cpp

Define Documentation

> **Warning:** doxygendefine: Cannot find define "FAC_INIT" in doxygen xml output for project "libecpint" from directory: ../../doxygen/xml/

Define H_START

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_api.hpp

Define Documentation

H_START(i, j, N)

returns the index for the ij-th second derivative for a system with N atoms

Define M_EULER

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Define Documentation

M_EULER

Define M_PI

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Define Documentation

M_PI

Define MAX_DFAC

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Define Documentation

MAX_DFAC

the maximum double factorial needed

Define MAX_FAC

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_mathutil.hpp

Define Documentation

MAX_FAC

the maximum factorial needed

Define N_INDEX

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_ecpint.hpp

Define Documentation

N_INDEX(l, m)

returns the index of the basis function with y,z angular momentum components l,m

Typedefs

Typedef libecpint::Heptuple

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Typedef Documentation

using `libecpint::`**`Heptuple`** = std::tuple<int, int, int, int, int, int, int>

Typedef libecpint::Pair

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Typedef Documentation

using `libecpint::`**`Pair`** = std::tuple<int, int>

Typedef libecpint::Quintuple

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Typedef Documentation

using `libecpint::`**`Quintuple`** = std::tuple<int, int, int, int, int>

Typedef libecpint::Triple

- Defined in file__Users_robertshaw_devfiles_libecpint_include_libecpint_multiarr.hpp

Typedef Documentation

using `libecpint::`**`Triple`** = std::tuple<int, int, int>

CHAPTER TWO

INDICES AND TABLES

- genindex
- modindex
- search

INDEX