POLARES (Radiative Corrections for Polarized Electron Nucleon Scattering)

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Namespace Documentation

3.1 POLARES Namespace Reference

Classes

- class Cross_Sections
- class Cuba_parameters
- class Final_State
- class Form_factors
- class Gamma_Loop
- class Input
- class Integrands
- class Interpolation
- class LoopTools
- class Melem
- class Melem_pol
- class Output
- class Parameters
- class PES
- class Rand
- class Scalar_Integrals
- class Virtual_Corrections

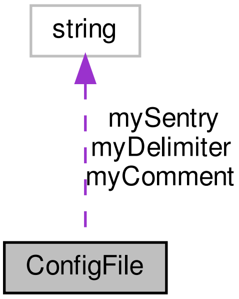
3.1.1 Detailed Description

Chapter 4

Class Documentation

4.1 ConfigFile Class Reference

Collaboration diagram for ConfigFile:



Classes

- struct file_not_found
- struct key_not_found

Public Member Functions

- **ConfigFile** (string filename, string delimiter="=", string comment="#", string sentry="EndConfigFile")
- template<class T >
  T **read** (const string &key) const
- template<class T >
  T **read** (const string &key, const T &value) const
- template<class T >
  bool **readInto** (T &var, const string &key) const

- template<class T >
  bool **readInto** (T &var, const string &key, const T &value) const
- template<class T >
  void **add** (string key, const T &value)
- void **remove** (const string &key)
- bool **keyExists** (const string &key) const
- string **getDelimiter** () const
- string **getComment** () const
- string **getSentry** () const
- string **setDelimiter** (const string &s)
- string **setComment** (const string &s)

Protected Types

- typedef std::map< string, string >::iterator **mapi**
- typedef std::map< string, string >::const_iterator **mapci**

Protected Member Functions

- template<>
  string **string_as_T** (const string &s)
- template<>
  bool **string_as_T** (const string &s)

Static Protected Member Functions

- template<class T >
  static string **T_as_string** (const T &t)
- template<class T >
  static T **string_as_T** (const string &s)
- static void **trim** (string &s)

Protected Attributes

- string **myDelimiter**
- string **myComment**
- string **mySentry**
- std::map< string, string > **myContents**

Friends

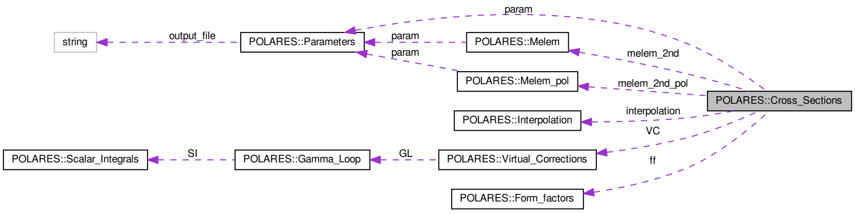
- std::ostream & **operator**<< (std::ostream &os, const ConfigFile &cf)
- std::istream & **operator**>> (std::istream &is, ConfigFile &cf)

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

- ReadConfigFile.h
- ReadConfigFile.cpp

4.2 POLARES::Cross_Sections Class Reference

Collaboration diagram for POLARES::Cross_Sections:



Public Member Functions

- int **set_param** (const Parameters ∗param, const Interpolation ∗interpolation)
- double **asymm_born_test** (const double Q2) const
- double **crsect_born** (const double Q2) const
- double **crsect_elastic** (const double Q2) const
- double **crsect_born_thl** (const double Q2) const
- double **crsect_elastic_thl** (const double Q2) const
- double **interf_born** (const double Q2) const
- double **interf_born_thl** (const double Q2) const
- double **interf_elastic** (const double Q2) const
- double **interf_elastic_thl** (const double Q2) const
- double **crsect_brems_1st** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **crsect_brems_1st_ps2** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **crsect_brems_1st_sg_diff** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **crsect_brems_1st_sp_vert** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **crsect_brems_2nd** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_add** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_add_phig1** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_sg_diff** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_sg_diff_interf** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_ps2** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_phig1** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **interf_brems_1st** (const double en1, const double thl, const double eg, const double thg) const
- double **interf_brems_1st_ps2** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **interf_brems_1st_test** (const double en1, const double thl, const double eg, const double thg) const
- double **interf_brems_2nd** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const

- double **crsect_brems_2nd_pol_add** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_2nd_pol_add_phig1** (const double en1, const double thl, const double eg, const double thg, const double phig, const double eg1, const double thg1, const double phig1) const
- double **crsect_brems_1st_hadr** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **crsect_brems_1st_hadr_interf** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **crsect_born_carbon** (const double Q2) const
- double **crsect_elastic_carbon** (const double Q2) const
- double **crsect_brems_1st_carbon** (const double en1, const double thl, const double eg, const double thg, const double phig) const
- double **interf_born_carbon** (const double Q2) const
- double **interf_elastic_carbon** (const double Q2) const
- double **interf_brems_1st_carbon** (const double en1, const double thl, const double eg, const double thg) const
- double **crsect_born_carbon_thl** (const double Q2) const
- double **crsect_elastic_carbon_thl** (const double Q2) const
- double **interf_born_carbon_thl** (const double Q2) const
- double **interf_elastic_carbon_thl** (const double Q2) const

Public Attributes

- Virtual_Corrections **VC**

Protected Attributes

- const Interpolation ∗ **interpolation**
- const Parameters ∗ **param**
- double **s**
- double **melem_interf**
- double **melem2**
- double **sigma_born**
- double **gpe**
- double **gpm**
- double **gpze**
- double **gpzm**
- double **gae**
- double **f1**
- double **f2**
- double **f1z**
- double **f2z**
- double **tau**
- double **eps**
- double **thl**
- double **l2**
- double **a**
- double **b**
- double **l1k**
- double **cospsi**
- double **x**
- double **phig**
- Form_factors **ff**

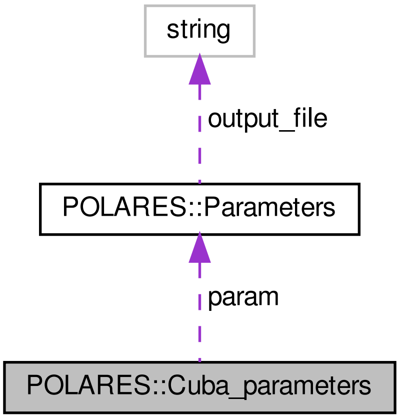
- Melem **melem_2nd**
- Melem_pol **melem_2nd_pol**
- double **gamma_loop**
- double **m**
- double **m2**
- double **m4**
- double **M**
- double **M2**
- double **M4**

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

- cross_sections.h
- cross_sections.cpp

4.3 POLARES::Cuba_parameters Class Reference

Collaboration diagram for POLARES::Cuba_parameters:



Public Member Functions

- void **set_param** (const Parameters ∗param)

Public Attributes

- char ∗ **STATEFILE_test**
- char ∗ **STATEFILE**
- int **GRIDNO_elastic**
- int **GRIDNO_brems**
- int **GRIDNO_brems_hadr**
- int **GRIDNO_brems_hadr_interf**
- int **GRIDNO_brems_1st**
- int **GRIDNO_brems_test**
- int **GRIDNO_brems_l1k**
- int **GRIDNO_brems_l2k**
- int **GRIDNO_brems_interf**
- int **GRIDNO_brems_2nd**
- int **GRIDNO_brems_2nd_l1k1**
- int **GRIDNO_brems_2nd_l1k2**
- int **GRIDNO_brems_2nd_l2k1**
- int **GRIDNO_brems_2nd_l2k2**
- int **GRIDNO_brems_interf_2nd**
- int **flags**
- int **flags_brems**
- double **SEED**
- int **MINEVAL**
- int **MAXEVAL_1st**
- int **MAXEVAL_2nd**
- int **MAXEVAL_2nd_add**
- double **EPSREL**
- int **no_cores**
- int ∗ **SPIN**
- int **NSTART**
- int **NINCREASE**
- int **NBATCH**
- int **NNEW**
- int **NMIN**
- double **FLATNESS**
- int **comp**
- int **neval**
- int **fail**
- int **nregions**

Static Public Attributes

- static const int **NDIM_ELASTIC** = 1
- static const int **NDIM_brems_1st** = 4
- static const int **NDIM_brems_2nd** = 7
- static const int **NDIM_brems_2nd_1diff** = 6
- static const int **NDIM_brems_2nd_2diff** = 5
- static const int **NDIM_brems_1st_1diff** = 3
- static const int **NDIM_brems_1st_2diff** = 2
- static const int **NCOMP** = 1
- static const int **NVEC** = 1
- static const int **KEY** = 0
- static const double **EPSABS** = 1e-200
- static const int **GRIDNO** = 0

Protected Attributes

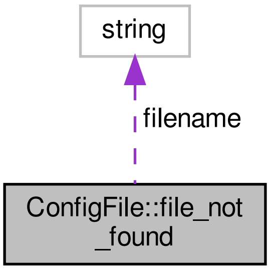
- const Parameters ∗ **param**

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

- cuba_param.h
- cuba_param.cpp

4.4 ConfigFile::file_not_found Struct Reference

Collaboration diagram for ConfigFile::file_not_found:



Public Member Functions

- **file_not_found** (const string &filename_=string())

Public Attributes

- string **filename**

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

- ReadConfigFile.h

4.5 POLARES::Final_State Class Reference

Public Attributes

- double **E**
- double **E_prime_l**
- double **theta_l**
- double **phi_l**
- double **E_p**
- double **theta_p**
- double **phi_p**
- double **E_gamma**
- double **theta_gamma**
- double **phi_gamma**
- double **E_gamma_prime**
- double **theta_gamma_prime**
- double **phi_gamma_prime**
- double **Q2**
- double **weight**
- double **avg_weight**
- double **sigma_diff**
- double **l_2** [4]
- double **k_1** [4]
- double **k_2** [4]
- double **p_2** [4]
- double **l_1** [4]
- double **p_1** [4]
- int **seed**
- int **event_no**
- int **event_type**
- int **failed_ev**

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

- IO_classes.h

4.6 POLARES::Form_factors Class Reference

Public Types

- enum **FF_Type** {
  FF_simple_dipole, **FF_pol_dipole_bern**, **FF_FW**, **FF_static_limit**,
  FF_user_defined, **FF_WRONG** }

Public Member Functions

- bool **change_flag** (const int flag)
- void **ffactp** (const double Q2, double &ge, double &gm) const
- void **ffactn** (const double Q2, double &gen, double &gmn) const
- void **sff** (const double Q2, double &gse, double &gsm) const
- void **ffgae** (const double Q2, double &gae) const
- void **ffz** (const double Q2, double &gpze, double &gpzm, const double sw2) const
- void **ffz** (const double Q2, double &gpze, double &gpzm, const double kappa, const double sw2) const
- double **ff_carbon12** (const double Q2) const
- double **ffz_carbon12** (const double Q2) const
- **Form_factors** (const int flag_ff)

Protected Attributes

- int **flag**

Static Protected Attributes

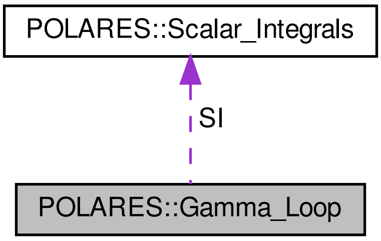
- static const double **pEa10** = 1.041
- static const double **pEa11** = 0.765
- static const double **pEa20** = -0.041
- static const double **pEa21** = 6.2
- static const double **pEab** = -0.23
- static const double **pEQb** = 0.07
- static const double **pEsigmab** = 0.27
- static const double **pMa10** = 1.002
- static const double **pMa11** = 0.749
- static const double **pMa20** = -0.002
- static const double **pMa21** = 6.0
- static const double **pMab** = -0.13
- static const double **pMQb** = 0.35
- static const double **pMsigmab** = 0.21
- static const double **ae** [8] = {-0.4980, 5.4592, -34.7281, 114.3173, -262.9808, 329.1395, -227.3306, 66.6980}
- static double const **am** [8] = {0.2472, -4.9123, 29.7509, -84.0430, 129.3256, -111.1068, 49.9753, -9.1659}
- static double const **amn** [10] = {-1.9147, 6.47767, -17.32918, 31.80021, -37.18707, 27.52359, -12.81713, 3.63457, -0.57277, 0.03843}
- static const double **a** = 2.28409
- static const double **b** = 4.41942
- static const double **aes** = 0.32267
- static const double **bes** = 4.686
- static const double **ams** = 0.044
- static const double **bms** = 0.93
- static const double **Gap** = -1.267
- static const double **M_a** = 1.014

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

- form_factors.h
- form_factors.cpp

4.7 POLARES::Gamma_Loop Class Reference

Collaboration diagram for POLARES::Gamma_Loop:



Public Member Functions

- long double se_off_shell (const long double Q2h, const long double Q2e, const long double l1k, const long double S1, const long double S2, const long double f1, const long double f2) const
- long double se_on_shell (const long double Q2h, const long double Q2e, const long double l1k, const long double S1, const long double S2, const long double f1, const long double f2) const
- long double vb_off_shell (const long double Q2h, const long double Q2e, const long double l1k, const long double S1, const long double S2, const long double f1, const long double f2) const
- long double vb_on_shell (const long double Q2h, const long double Q2e, const long double l1k, const long double S1, const long double S2, const long double f1, const long double f2) const
- int **set_param** (const Parameters ∗param)

Public Attributes

- Scalar_Integrals **SI**

4.7.1 Member Function Documentation

4.7.1.1 se_off_shell()

```
long double Gamma_Loop::se_off_shell (
             const long double Q2h,
             const long double Q2e,
             const long double l1k,
             const long double S1,
             const long double S2,
             const long double f1,
             const long double f2 ) const
```

Self-energy with a hard-photon emitted from the off-shell line

4.7.1.2 se_on_shell()

```
long double Gamma_Loop::se_on_shell (
             const long double Q2h,
             const long double Q2e,
             const long double l1k,
             const long double S1,
             const long double S2,
             const long double f1,
             const long double f2 ) const
```

Self-energy with a hard-photon emitted from the on-shell line

4.7.1.3 vb_off_shell()

```
long double Gamma_Loop::vb_off_shell (
             const long double Q2h,
             const long double Q2e,
             const long double l1k,
             const long double S1,
             const long double S2,
             const long double f1,
             const long double f2 ) const
```

Vertex correction with a hard-photon emitted from the off-shell line

4.7.1.4 vb_on_shell()

```
long double Gamma_Loop::vb_on_shell (
             const long double Q2h,
             const long double Q2e,
             const long double l1k,
             const long double S1,
             const long double S2,
             const long double f1,
             const long double f2 ) const
```

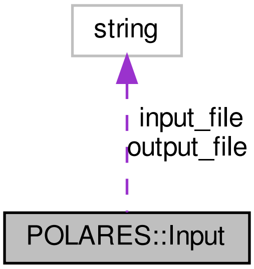
Vertex correction with a hard-photon emitted from the on-shell line

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

- gamma_loop.h
- gamma_loop.cpp

4.8 POLARES::Input Class Reference

Collaboration diagram for POLARES::Input:



Public Types

- enum **Flags** {
  vac_pol, **order**, **brems**, **asymmetry**,
  LO, **kappa_weak**, **cuts_born**, **ps**,
  form_factors, **tpe**, **echo_input**, **lepton**,
  brems_add, **int_method**, **GL**, **PS**,
  brems_hadr, **target**, **int_output** }

Public Attributes

- double **thl_min**
- double **thl_max**
- double **polarization**
- double **E_prime_min**
- double **E_prime_max**
- double **Delta**
- double **Delta1**
- double **thl_deg**
- double **thg_deg**
- double **E**
- double **E_min**
- double **E_max**
- double **Delta_E**
- double **thg_min**
- double **thg_max**
- double **E_gamma_max**
- double **Q2min**
- double **Q2max**
- double **Delta_eps**
- double **mu_dim**
- double **lambda**

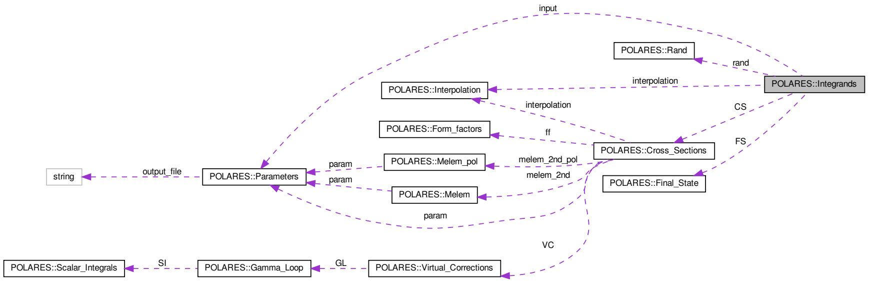
- double **sw2**
- double **flatness**
- double **seed**
- double **epsrel**
- int **no_cores**
- int **no_eval_2nd**
- int **no_eval_1st**
- int **no_eval_gamma_loop**
- int **no_eval_2nd_add**
- int **no_min_eval**
- int **nstart**
- int **nincrease**
- int **nbatch**
- int **nnew**
- int **nmin**
- int **flag** [30]
- std::string **input_file**
- std::string **output_file**

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

- IO_classes.h

4.9 POLARES::Integrands Class Reference

Collaboration diagram for POLARES::Integrands:



Public Member Functions

- void **set_input** (const Parameters ∗input, const Interpolation ∗interpolation, Final_State ∗FS)
- void **set_input** (const Parameters ∗input, const Interpolation ∗interpolation)
- int **integrand_born** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_elastic** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_born** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_elastic** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_hadr** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_hadr_interf** (const double xx[ ], double ff[ ], const double weight[ ]) const

- int **integrand_brems_1st_sp_vert** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_l1k** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_l2k** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_2diff_v1** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_2diff_v2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_ps2_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_ps2_3diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_ps2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_sg_diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_1st** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_1st_ps2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_thl_ps2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_hadr_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_hadr_interf_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_1st_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add_l1k1** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add_l1k2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add_l2k1** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add_l2k2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l1k1** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l1k2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l2k1** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l2k2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_sg_diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_sg_diff_interf** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_Q2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_add_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_1diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l1k1_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l1k2_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l2k1_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_l2k2_2diff** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_test** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_pol_add** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_2nd_pol_add_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_2nd** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_2nd_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_shiftQ2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_shiftQ2_ps2** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_shiftQ2_2nd** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_born_carbon** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_elastic_carbon** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_born_carbon** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_elastic_carbon** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_carbon** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_1st_carbon** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_brems_1st_carbon_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const
- int **integrand_interf_brems_1st_carbon_thl** (const double xx[ ], double ff[ ], const double weight[ ]) const

Static Public Member Functions

- static int **cuba_integrand_born** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_elastic** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_born** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_elastic** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_hadr** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_hadr_interf** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_sp_vert** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_l1k** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_l2k** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_2diff_v1** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_2diff_v2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_ps2_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_ps2_3diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_ps2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_sg_diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add_l1k1** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add_l1k2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add_l2k1** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add_l2k2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l1k1** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l1k2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l2k1** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l2k2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_sg_diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)

- static int **cuba_integrand_brems_2nd_sg_diff_interf** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_Q2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_add_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_1diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l1k1_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l1k2_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l2k1_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_l2k2_2diff** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_test** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_1st** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_1st_ps2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_2nd** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_pol_add** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_2nd_pol_add_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_2nd_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_shiftQ2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_shiftQ2_ps2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_shiftQ2_2nd** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_thl_ps2** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_1st_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_hadr_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_hadr_interf_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_born_carbon** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_elastic_carbon** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)

- static int **cuba_integrand_interf_born_carbon** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_elastic_carbon** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_carbon** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_1st_carbon** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_brems_1st_carbon_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)
- static int **cuba_integrand_interf_brems_1st_carbon_thl** (const int ∗ndim, const double xx[ ], const int ∗ncomp, double ff[ ], void ∗userdata, const int ∗nvec, const int ∗core, const double weight[ ], const int ∗iter)

Public Attributes

- Cross_Sections **CS**

Protected Attributes

- const Parameters ∗ **input**
- const Interpolation ∗ **interpolation**
- Final_State ∗ **FS**
- Rand **rand**
- double **Q2**
- double **Jacobian**
- double **max_wgt**
- double **Q2min**
- double **Q2max**
- int **events_no**
- int **acc_events**
- double **a**
- double **b**
- double **c**
- double **d**
- double **f**
- double **n2**
- double **l1**
- double **l2**
- double **a1**
- double **a2**
- double **y**
- double **Jacobian_eg**
- double **Jacobian_eg1**
- double **thgmin**
- double **thgmax**
- double **egmin**
- double **egmax**
- double **thlmin**
- double **eg1min**
- double **eg1max**
- double **thg1min**
- double **thg1max**
- double **phig1min**

- double **phig1max**
- double **phig1min_cos**
- double **phig1max_cos**
- double **thl**
- double **thg**
- double **phig**
- double **en1**
- double **eg**
- double **eg1**
- double **thg1**
- double **phig1**
- double **m**
- double **m2**
- double **M**
- double **M2**

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

- integrands.h
- integrands.cpp

4.10 POLARES::Interpolation Class Reference

Public Member Functions

- void **init_d_vac_hadr** (const std::string &filename)
- void **init_tpe_Tomalak** (const std::string &filename)
- void **init_tpe** (const std::string &filename)
- double **d_vac_hadr** (const double Q2) const
- double **tpe_Tomalak** (const double Q2) const
- double **tpe** (const double Q2, const double eps) const
- **Interpolation** (const Interpolation &I)

Protected Attributes

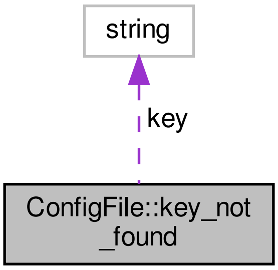
- void $*$ **gsl_interp_accel_ptr**
- void $*$ **gsl_interp_accel_ptr_tpe_Tomalak**
- void $*$ **gsl_spline_ptr**
- void $*$ **gsl_spline_ptr_tpe_Tomalak**
- void $*$ **gsl_spline2d_ptr**
- void $*$ **gsl_xacc_ptr**
- void $*$ **gsl_yacc_ptr**

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

- interpolation.h
- interpolation.cpp

4.11 ConfigFile::key_not_found Struct Reference

Collaboration diagram for ConfigFile::key_not_found:



Public Member Functions

- **key_not_found** (const string &key_=string())

Public Attributes

- string **key**

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

- ReadConfigFile.h

4.12 POLARES::LoopTools Class Reference

Public Member Functions

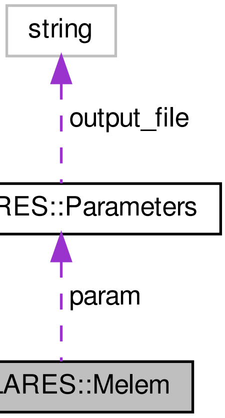
- double **C0_looptools** (const double p1, const double p2, const double p3, const double m1, const double m2, double const m3) const
- double **D0_looptools** (const double p1, const double p2, const double p3, const double p4, const double Q12, const double Q23, const double m1, const double m2, double const m3, const double m4) const
- void **set_Delta_eps** (const double Delta_eps)
- void **set_mudim2** (const double mu_dim2)
- void **set_lambda2** (const double lambda2)

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

- looptools_interface.h
- looptools_interface.cpp

4.13 POLARES::Melem Class Reference

Collaboration diagram for POLARES::Melem:



Public Member Functions

- int **set_param** (const Parameters ∗param)
- double **melem2** (const double l1k1, const double l1k2, const double k1k2, const double Q2e, const double Q2h, const double Q2k, const double S, const double Sk, const double Sq2, const double f1, const double f2) const
- double **melem2_add** (const double l1k1, const double l1k2, const double k1k2, const double Q2e, const double Q2h, const double Q2k, const double S, const double Sk, const double Sq2, const double f1, const double f2) const
- double **melem2_add_interf** (const double l1k1, const double l1k2, const double l2k1, const double l2k2, const double Q2e, const double Q2h, const double S, const double Sk, const double Sq2, const double f1, const double f2) const
- double **melem2_sg_diff** (const double l1k1, const double l1k2, const double k1k2, const double Q2e, const double Q2h, const double Q2k, const double S, const double Sk, const double Sq2, const double f1, const double f2) const
- double **melem2_sg_diff_interf** (const double l1k1, const double l1k2, const double l2k1, const double l2k2, const double Q2e, const double Q2h, const double S, const double Sk, const double Sq2, const double f1, const double f2) const

Protected Attributes

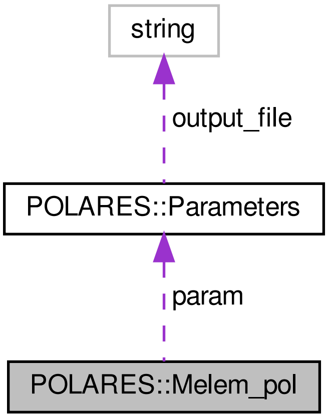
- const Parameters ∗ **param**
- double **m**
- double **m2**
- double **m4**
- double **m6**
- double **M**
- double **M2**

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

- melem.h
- melem.cpp

4.14 POLARES::Melem_pol Class Reference

Collaboration diagram for POLARES::Melem_pol:



Public Member Functions

- int **set_param** (const Parameters ∗param)
- double **melem_interf** (const double l1k1, const double l1k2, const double k1k2, const double Q2e, const double Q2h, const double Q2k, const double S, const double Sk, const double Sq2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **melem2_pol_add** (const double l1k1, const double l1k2, const double l2k1, const double l2k2, const double Q2e, const double Q2h, const double S, const double Sk, const double Sq2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **melem2_pol_add_interf_g** (const double l1k1, const double l1k2, const double l2k1, const double l2k2, const double Q2e, const double Q2h, const double S, const double Sk, const double Sq2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **melem2_pol_add_interf_Z** (const double l1k1, const double l1k2, const double l2k1, const double l2k2, const double Q2e, const double Q2h, const double S, const double Sk, const double Sq2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const

Protected Attributes

- const Parameters ∗ **param**
- double **m**
- double **m2**
- double **m4**
- double **m6**
- double **M**
- double **M2**

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

- melem_pol.h
- melem_pol.cpp

4.15 POLARES::Output Class Reference

Public Attributes

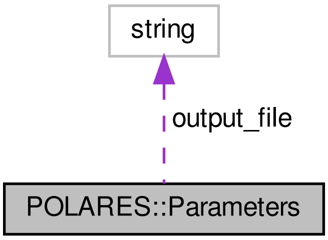
- double **sigma_unpol_born**
- double **sigma_pol_born**
- double **sigma_unpol_elastic_1st**
- double **sigma_unpol_elastic_2nd**
- double **sigma_pol_elastic_1st**
- double **sigma_pol_elastic_2nd**
- double **sigma_unpol_inelastic_1st**
- double **sigma_unpol_inelastic_2nd**
- double **sigma_unpol_inelastic_loop**
- double **sigma_pol_inelastic_loop**
- double **sigma_pol_inelastic_1st**
- double **sigma_pol_inelastic_2nd**
- double **sigma_unpol_1st**
- double **sigma_pol_1st**
- double **sigma_unpol_2nd**
- double **sigma_pol_2nd**
- double **sigma_born**
- double **sigma_1st**
- double **sigma_2nd**
- double **asymm_born**
- double **asymm_1st**
- double **asymm_2nd**
- double **rel_asymm_1st**
- double **rel_asymm_2nd**
- double **sigma_unpol_2nd_add**
- double **sigma_pol_2nd_add**
- double **sigma_unpol_inelastic_1st_hadr**
- double **sigma_unpol_inelastic_1st_hadr_interf**
- std::vector< double > **sigma_unpol_1st_vect**
- std::vector< double > **sigma_pol_1st_vect**
- std::vector< double > **ev_brems_1st**
- std::vector< double > **sigma_unpol_1st_elastic_vect**
- std::vector< double > **sigma_unpol_1st_inelastic_vect**
- std::vector< double > **sigma_unpol_2nd_vect**
- std::vector< double > **sigma_1st_vect**
- std::vector< double > **sigma_2nd_vect**
- std::vector< double > **sigma_pol_2nd_vect**
- std::vector< double > **ev_brems_2nd**
- std::vector< double > **ev_brems_2nd_l1k1**
- std::vector< double > **ev_brems_2nd_l1k2**
- std::vector< double > **ev_brems_2nd_l2k1**
- std::vector< double > **ev_brems_2nd_l2k2**
- double **shiftQ2**
- double **rel_shiftQ2**
- double **Q2**

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

- IO_classes.h

4.16 POLARES::Parameters Class Reference

Collaboration diagram for POLARES::Parameters:



Public Types

- enum **Flags** {
  asymmetry, **cuts_born**, **polarization**, **ps**,
  form_factors, **vac_pol**, **tpe**, **echo_input**,
  lepton, **brems**, **brems_add**, **shiftQ2_2nd**,
  LO, **order**, **int_method**, **kappa_weak**,
  GL, **PS**, **brems_hadr**, **target**,
  int_output }
- enum **Cuts** {
  E, **E_prime**, **theta_l**, **theta_l_deg**,
  cos_thl, **E_gamma**, **theta_gamma**, **cos_thg**,
  theta_gamma_deg, **phi_gamma**, **Q2_elastic**, **E_gamma_prime** }

Public Member Functions

- int **read_input** (const Input &input)
- int **final_param** (const Input &input)
- int **set_thl** (const double thl_deg)

Public Attributes

- double **SEED**
- int **MINEVAL**
- int **MAXEVAL_1st**
- int **maxeval_1st_aux**
- int **MAXEVAL_gamma_loop**
- int **MAXEVAL_2nd**
- int **MAXEVAL_2nd_add**
- double **EPSREL**
- int **no_cores**
- int **GRIDNO_elastic**
- int **GRIDNO_brems**

- int **GRIDNO_brems_hadr**
- int **GRIDNO_brems_hadr_interf**
- int **GRIDNO_brems_1st**
- int **GRIDNO_brems_test**
- int **GRIDNO_brems_l1k**
- int **GRIDNO_brems_l2k**
- int **GRIDNO_brems_interf**
- int **GRIDNO_brems_2nd**
- int **GRIDNO_brems_2nd_l1k1**
- int **GRIDNO_brems_2nd_l1k2**
- int **GRIDNO_brems_2nd_l2k1**
- int **GRIDNO_brems_2nd_l2k2**
- int **GRIDNO_brems_interf_2nd**
- int **NSTART**
- int **NINCREASE**
- int **NBATCH**
- int **NNEW**
- int **NMIN**
- double **FLATNESS**
- int **flag** [30]
- double **min** [20]
- double **max** [20]
- double **P**
- double **m**
- double **m2**
- double **M**
- double **M2**
- double **sw2**
- double **Z_lepton**
- double **Z_target**
- double **en**
- double **l1**
- double **thl**
- double **Q2**
- double **eps**
- double **thg**
- double **aux**
- double **Delta_E**
- double **Delta_eps**
- double **mu_dim**
- double **lambda**
- std::string **output_file**

Protected Attributes

- int **check_input**

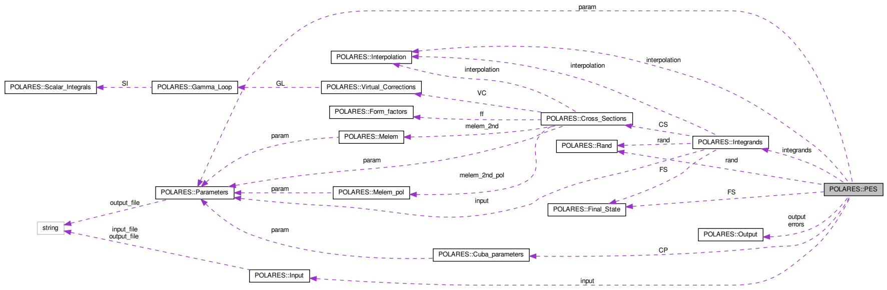
The documentation for this class was generated from the following files:

- parameters.h
- parameters.cpp

4.17 POLARES::PES Class Reference

```
#include <POLARES.h>
```

Collaboration diagram for POLARES::PES:



Public Member Functions

- PES (const Input &input)
- PES ()
- bool change_energy_initialization (const double E)
- bool change_energy_events (const double E)
- bool set_child_process (const int child_process)
- int initialization ()
- int events ()
- int shiftQ2 (const double thl_deg)
- int sigma_diff_Omega_l (const double thl_deg)
- double delta (const double thl_deg)
- double running_sw2 (const double Q2)
- void set_input (const Input &input)
- double get_total_unpol_cross_section (const double E)
- double get_E ()

Public Attributes

- Output output
- Output errors
- Final_State FS

Protected Member Functions

- **PES** (const PES &)
- PES & **operator=** (const PES &)
- void set_final_state ()
- void **write_output** ()

Protected Attributes

- Parameters param
- Interpolation interpolation
- Input input
- Rand rand
- Cuba_parameters CP
- Integrands integrands
- double m
- double **M**
- double **m2**
- double **M2**
- bool **seed_is_set**

4.17.1 Detailed Description

main class

4.17.2 Constructor & Destructor Documentation

4.17.2.1 PES() [1/2]

```
POLARES::PES::PES (
             const Input & input )
```

Constructor that can receive external user-defined input sets the input

Parameters

input	

4.17.2.2 PES() [2/2]

```
PES::PES ( )
```

Standard Constructor sets the input

4.17.3 Member Function Documentation

4.17.3.1 change_energy_events()

```
bool PES::change_energy_events (
            const double E )
```

Function that can be used to change the energy for generating events

Parameters

E	(energy of the incoming electron)

Returns

true if the value was initialized

4.17.3.2 change_energy_initialization()

```
bool PES::change_energy_initialization (
            const double E )
```

Function that can be used to change the energy before the initialization

Parameters

E	(energy of the incoming electron)

Returns

true if the value can be used to generate events

4.17.3.3 delta()

```
double PES::delta (
            const double thl_deg )
```

Function that calculates the total (virtual + real) first order radiative corrections

Returns

results and errors

4.17.3.4 events()

```
int PES::events ( )
```

Function that generates events. Can be used only after the initialization was performed. Warning! The current version works only with energies between 20-200 MeV with an increment of 0.1 MeV

Returns

phase_space

4.17.3.5 get_E()

```
double PES::get_E ( )
```

Function that echoes the energy provided by the user

Returns

sigma_unpol_vect

4.17.3.6 get_total_unpol_cross_section()

```
double PES::get_total_unpol_cross_section (
             const double E )
```

Function that gives the total cross section for a given energy

Returns

sigma_unpol_vect

4.17.3.7 initialization()

```
int PES::initialization ( )
```

Function that can be used for the initialization of the grid in case events are generated or simply to obtain the cross sections or asymmetries

Returns

results and errors

4.17.3.8 running_sw2()

```
double PES::running_sw2 (
             const double Q2 )
```

Function that calculates the running of the weak mixing angle

Returns

results

4.17.3.9 set_child_process()

```
bool PES::set_child_process (
             const int child_process )
```

Function for changing the events seed in case of multi-core usage

Parameters

child_process	

Returns

true if the value is different from other seeds

4.17.3.10 set_final_state()

```
void PES::set_final_state ( )  [protected]
```

A function that sets the final values for the final state particles

4.17.3.11 set_input()

```
void PES::set_input (
             const Input & input )
```

Function that can be used to replace the input from "POLARES.in"

Returns

void

4.17.3.12 shiftQ2()

```
int PES::shiftQ2 (
             const double thl_deg )
```

Function that calculates the shift in Q2 due to first order hard-photon bremsstrahlung for a given scattering angle theta _l

Returns

results and errors

4.17.3.13 sigma_diff_Omega_l()

```
int PES::sigma_diff_Omega_l (
             const double thl_deg )
```

Function that calculates the one-fold differential cross-section and asymmetry in respect to the scattering angle of the final electron

Returns

results and errors

4.17.4 Member Data Documentation

4.17.4.1 CP

```
Cuba_parameters POLARES::PES::CP  [protected]
```

Cuba related parameters

4.17.4.2 errors

```
Output POLARES::PES::errors
```

A class in which the uncertainties of the numerical integration are placed

4.17.4.3 FS

```
Final_State POLARES::PES::FS
```

A class in which all the information about the final state particles is stored when events are generated

4.17.4.4 input

```
Input POLARES::PES::input  [protected]
```

Class for storing user defined input

4.17.4.5 integrands

```
Integrands POLARES::PES::integrands  [protected]
```

Class that contains all the integrands

4.17.4.6 interpolation

```
Interpolation POLARES::PES::interpolation  [protected]
```

Interpolation of vacuum polarization and two-photon exchange corrections

4.17.4.7 m

```
double POLARES::PES::m  [mutable], [protected]
```

Lepton and target particle masses

4.17.4.8 output

```
Output POLARES::PES::output
```

A class in which the results of the numerical integration are placed

4.17.4.9 param

```
Parameters POLARES::PES::param  [protected]
```

Storing for all the required parameters

4.17.4.10 rand

```
Rand POLARES::PES::rand  [protected]
```

GSL random number generator

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

- POLARES.h
- POLARES.cpp

4.18 POLARES::Rand Class Reference

Public Types

- enum **GSL_RAND_TYPE** {
  GSL_RAND_TYPE_mt19937, **GSL_RAND_TYPE_ranlxs0**, **GSL_RAND_TYPE_ranlxs1**, **GSL_RAND_TYPE_ranlxs2**,
  GSL_RAND_TYPE_ranlxd1, **GSL_RAND_TYPE_ranlxd2**, **GSL_RAND_TYPE_ranlux**, **GSL_RAND_TYPE_ranlux389**,
  GSL_RAND_TYPE_cmrg, **GSL_RAND_TYPE_mrg**, **GSL_RAND_TYPE_taus**, **GSL_RAND_TYPE_taus2**,
  GSL_RAND_TYPE_gfsr4 }

Public Member Functions

- unsigned int **get_seed** () const
- GSL_RAND_TYPE **get_type** () const
- void **change_seed** (unsigned long int in_seed)
- **Rand** (const unsigned long int seed=time(NULL), const GSL_RAND_TYPE type=GSL_RAND_TYPE_ranlux)
- **Rand** (const Rand &R)
- Rand & **operator=** (const Rand &R)
- double **uniform** () const
- double **uniform** (const double min, const double max) const
- double **uniform_pos** () const
- double **uniform_pos** (const double min, const double max) const
- double **operator()** () const
- double **operator()** (const double min, const double max) const
- long int **uniform_int** (const long int max) const
- std::ostream & **to_stream** (std::ostream &os) const

Protected Member Functions

- void **_init** ()
- void **_exit** ()

Protected Attributes

- unsigned long int **seed**
- GSL_RAND_TYPE **type**
- unsigned int ∗ **stack**
- const void ∗ **gsl_T**
- void ∗ **gsl_G**

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

- gsl_rand.h
- gsl_rand.cpp

4.19 POLARES::Scalar_Integrals Class Reference

Public Member Functions

- long double A0_m (const long double m2) const
- long double B0_0mm (const long double m2) const
- long double B0_m0m (const long double m2) const
- long double B0_00m (const long double m2) const
- long double B0_qmm (const long double Q2e, const long double m2) const
- long double B0_M0m (const long double M2, const long double m2) const
- long double C0_mmqm0m (const long double Q2e, const long double m2) const
- long double C0_mm0m0m (const long double m) const
- long double C0_m0M0mm (const long double M2, const long double m2) const
- long double C0_0qQmmm (const long double Q2e, const long double Q2h, const long double m2) const
- long double C0_mMQm0m (const long double M2, const long double Q2h, const long double m2) const
- long double D0_mm0QqMm0mm (const long double Q2h, const long double Q2e, const long double M2, const long double m2) const
- int **set_param** (const Parameters ∗param)

4.19.1 Member Function Documentation

4.19.1.1 A0_m()

```
long double Scalar_Integrals::A0_m (
             const long double m2 ) const
```

1-point function

Returns

A0(m2)

4.19.1.2 B0_00m()

```
long double Scalar_Integrals::B0_00m (
             const long double m2 ) const
```

2-point function

Returns

B0(0,0,m2)

4.19.1.3 B0_0mm()

```
long double Scalar_Integrals::B0_0mm (
             const long double m2 ) const
```

2-point function

Returns

B0(0,m2,m2)

4.19.1.4 B0_m0m()

```
long double Scalar_Integrals::B0_m0m (
             const long double m2 ) const
```

2-point function

Returns

B0(m2,0,m2)

4.19.1.5 B0_M0m()

```
long double Scalar_Integrals::B0_M0m (
             const long double M2,
             const long double m2 ) const
```

2-point function

Returns

B0(M2,0,m2)

4.19.1.6 B0_qmm()

```
long double Scalar_Integrals::B0_qmm (
             const long double Q2e,
             const long double m2 ) const
```

2-point function

Returns

B0(-Q2e,m2,m2)

4.19.1.7 C0_0qQmmm()

```
long double Scalar_Integrals::C0_0qQmmm (
             const long double Q2e,
             const long double Q2h,
             const long double m2 ) const
```

3-point function

Returns

C0(0,-Q2e,-Q2h,m2,m2,m2)

4.19.1.8 C0_m0M0mm()

```
long double Scalar_Integrals::C0_m0M0mm (
             const long double M2,
             const long double m2 ) const
```

3-point function

Returns

C0(m2,0,M2,0,m2,m2)

4.19.1.9 C0_mm0m0m()

```
long double Scalar_Integrals::C0_mm0m0m (
             const long double m ) const
```

IR divergent 3-point function regularized with a photon mass lambda

Returns

C0(m2,m2,0,m2,0,m2)

4.19.1.10 C0_mmqm0m()

```
long double Scalar_Integrals::C0_mmqm0m (
             const long double Q2e,
             const long double m2 ) const
```

IR divergent 3-point function regularized with a photon mass lambda

Returns

C0(m2,m2,-Q2e,m2,0,m2)

4.19.1.11 C0_mMQm0m()

```
long double Scalar_Integrals::C0_mMQm0m (
            const long double M2,
            const long double Q2h,
            const long double m2 ) const
```

3-point function calculated by LoopTools to be used only if LoopTools is included

Returns

C0(m2,M2,-Q2h,m2,0,m2)

4.19.1.12 D0_mm0QqMm0mm()

```
long double Scalar_Integrals::D0_mm0QqMm0mm (
            const long double Q2h,
            const long double Q2e,
            const long double M2,
            const long double m2 ) const
```

IR divergent 4-point function calculated by LoopTools to be used only if LoopTools is included regularized with a photon mass lambda

Returns

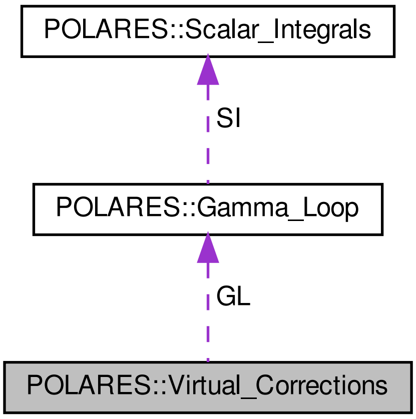
D0(m2,m2,0,-Q2h,-Q2e,M2,m2,0,m2,m2)

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

- scalar_integrals.h
- scalar_integrals.cpp

4.20 POLARES::Virtual_Corrections Class Reference

Collaboration diagram for POLARES::Virtual_Corrections:



Public Member Functions

- double **d_vac_1st** (const double Q2, const double m_lepton) const
- double **d_vac_2nd** (const double Q2, const double m_lepton) const
- double **d_box_Feshbach** (const double Q2) const
- double **d_box_MTj** (const double Q2) const
- double **d_vert** (const double Q2, const double f1, const double f2) const
- double **d_vert_pol** (const double Q2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **d_vert_pol_g** (const double Q2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **d_vert_pol_Z** (const double Q2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **d_vert_pol_quad_gZ** (const double Q2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double **d_vert_quad** (const double Q2, const double f1, const double f2) const
- double **d_vert_carbon** (const double Q2) const
- double **d_brems_ee** (const double Q2) const
- double **d_brems_hadr** (const double Q2) const
- double **d_brems_ee_test** (const double Q2) const
- long double **d_gamma_loop** (const long double Q2h, const long double Q2e, const long double l1k, const long double S1, const long double S2, const long double f1, const long double f2) const
- long double **d_gamma_loop_pol** (const long double Q2h, const long double Q2e, const long double l1k, const long double S1, const long double S2, const long double f1, const long double f2) const
- double **d_2nd_total** (const double Q2, const double f1, const double f2) const
- double **d_2nd_pol_total** (const double Q2, const double f1, const double f2, const double f1z, const double f2z, const double gae, const double ga, const double gv) const
- double kappa_weak (const double Q2) const
- int **set_param** (const Parameters ∗param)

Public Attributes

- Gamma_Loop **GL**

4.20.1 Member Function Documentation

4.20.1.1 kappa_weak()

```
double Virtual_Corrections::kappa_weak (
             const double Q2 ) const
```

weak corrections arXiv : 1107.4683, formulas extracted from Jegerlehner's code alphaQED using effective quark masses, A. Weber, H. Spiesberger

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

- virtual_corrections.h
- virtual_corrections.cpp