Kinematic Factor

1.0

Generated by Doxygen 1.8.15

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

Chapter 2

Data Type Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Module Documentation

5.1 IrrepName Namespace Reference

Functions

- std::vector< std::string > getIrrep (int &twoJ, int &P, string &lg)
- int irrepRows (string &irrep)

5.1.1 Function Documentation

5.1.1.1 getIrrep()

```
std::vector< std::string > IrrepName::getIrrep (
            int & twoJ,
            int & P,
            string & lg )
```

Here is the caller graph for this function:

5.1.1.2 irrepRows()

```
int IrrepName::irrepRows (
            string & irrep )
```

Here is the caller graph for this function:

5.2 iter Namespace Reference

Functions

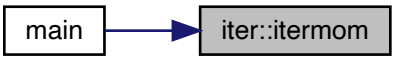
- std::vector< Vector3d > itermom (double max_mom)

5.2.1 Function Documentation

5.2.1.1 itermom()

```
std::vector< Vector3d > iter::itermom (
            double max_mom )
```

Here is the caller graph for this function:



5.3 KFactorEnv Namespace Reference

Functions

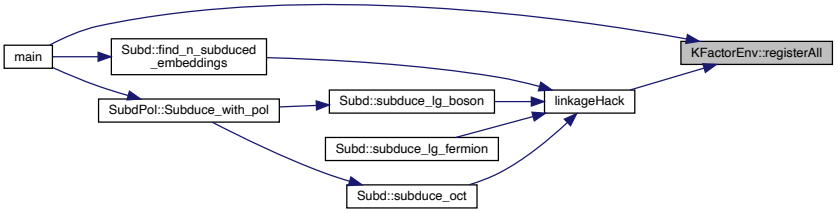
- bool registerAll ()

5.3.1 Function Documentation

5.3.1.1 registerAll()

```
bool KFactorEnv::registerAll ( )
```

Here is the caller graph for this function:



5.4 KfUt Namespace Reference

Data Structures

- class ToArray

Functions

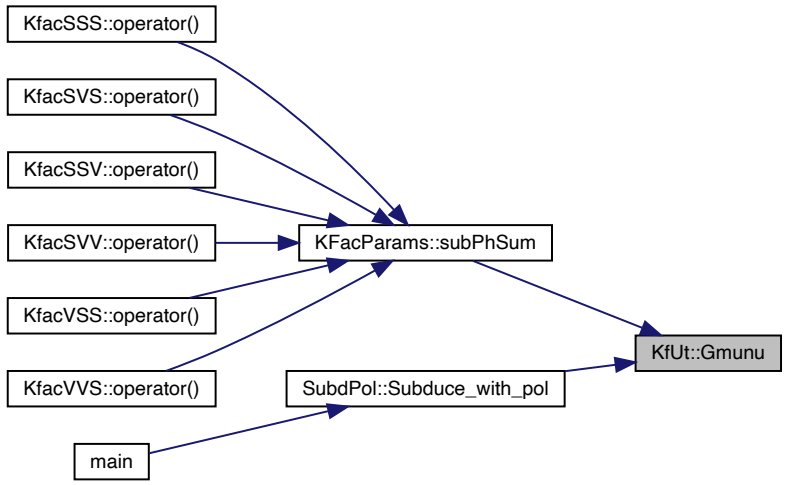
- double truncate (double num, int precision)
- Eigen::MatrixXcd Gmunu ()

5.4.1 Function Documentation

5.4.1.1 Gmunu()

```
Eigen::MatrixXcd KfUt::Gmunu ( )
```

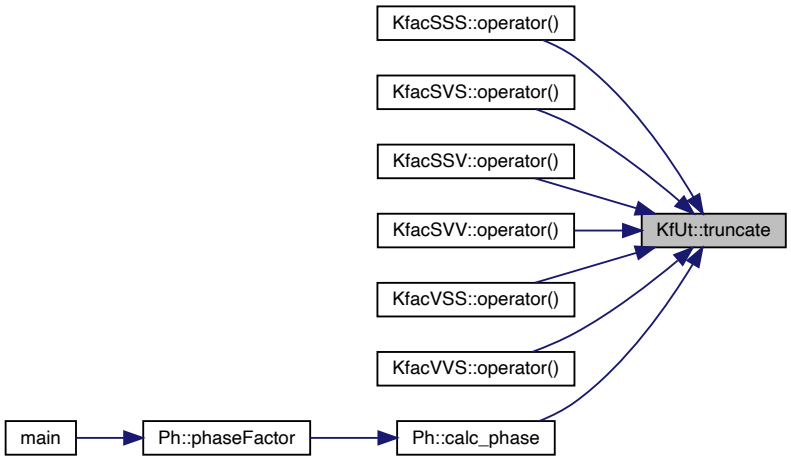
Here is the caller graph for this function:



5.4.1.2 truncate()

```
double KfUt::truncate (
            double num,
            int precision )
```

Here is the caller graph for this function:



5.5 LevCiv Namespace Reference

Functions

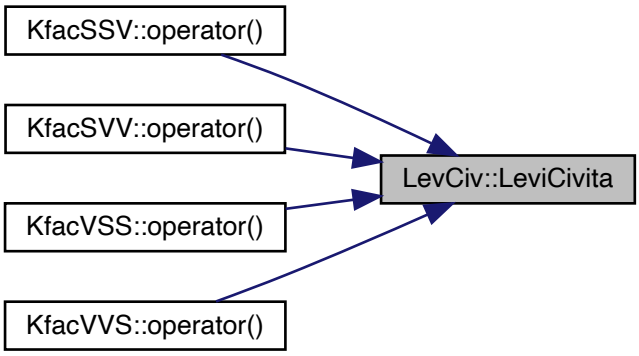
- double LeviCivita (int arr[ ], int n)

5.5.1 Function Documentation

5.5.1.1 LeviCivita()

```
double LevCiv::LeviCivita (
            int arr[],
            int n )
```

Here is the caller graph for this function:



5.6 LittleGrp Namespace Reference

Functions

- string generateLittleGroup (Eigen::Vector3d &mom_)
- std::vector< double > refAngles (Eigen::Vector3d mom1)

5.6.1 Function Documentation

5.6.1.1 generateLittleGroup()

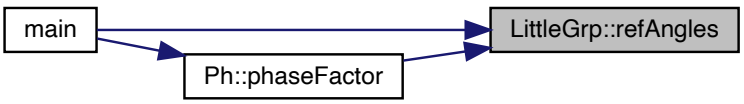
```
string LittleGrp::generateLittleGroup (
            Eigen::Vector3d & mom_ )
```

Here is the caller graph for this function:

5.6.1.2 refAngles()

```
std::vector< double > LittleGrp::refAngles (
            Eigen::Vector3d mom1 )
```

Here is the caller graph for this function:



5.7 naming Namespace Reference

Functions

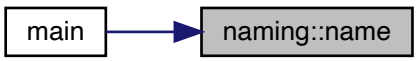
- string name (int npt, Ph::tripKey two_abs_lam, Vector3d mom1, Vector3d mom_curr, Vector3d mom3, irrep_label rep1, irrep_label rep_curr, irrep_label rep3, string LG1, string LG_curr, string LG3, string lev1, string lev3)

5.7.1 Function Documentation

5.7.1.1 name()

```
string naming::name (
            int npt,
            Ph::tripKey two_abs_lam,
            Vector3d mom1,
            Vector3d mom_curr,
            Vector3d mom3,
            irrep_label rep1,
            irrep_label rep_curr,
            irrep_label rep3,
            string LG1,
            string LG_curr,
            string LG3,
            string lev1,
            string lev3 )
```

Here is the caller graph for this function:



5.8 Ph Namespace Reference

Data Structures

- struct phChars

Typedefs

- typedef std::tuple< int, int, int > tripKey

Functions

- Ph::phChars phaseFactor (int twoJ1, int twoJ2, int twoJCurr, Eigen::Vector3d mom1, Eigen::Vector3d mom2, bool compute)
- std::complex< double > comp_Wigner_d (int twoJ, int twolam1, int twolam2, double a1, double b1, double c1, double a2, double b2, double c2, int n)
- map< Ph::tripKey, complex< double > > calc_phase (int twoJ1, int twoJ2, int twoJCurr, double mom1_sq, double mom2_sq, double mom_curr_sq, vector< double > r_mom1, vector< double > r_n_mom1, vector< double > r_mom2, vector< double > r2, vector< double > r_mom_curr, vector< double > r_n_mom_curr)
- map< Ph::tripKey, complex< double > > cnst_phase (int twoJ1, int twoJ2, int twoJCurr)

5.8.1 Typedef Documentation

5.8.1.1 tripKey

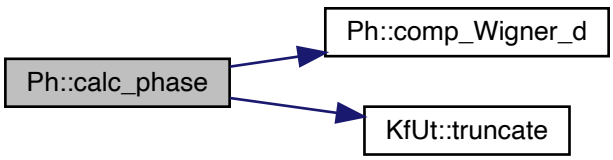
```
typedef std::tuple<int, int, int> Ph::tripKey
```

5.8.2 Function Documentation

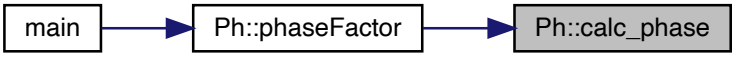
5.8.2.1 calc_phase()

```
map< Ph::tripKey, complex< double > > Ph::calc_phase (
            int twoJ1,
            int twoJ2,
            int twoJCurr,
            double mom1_sq,
            double mom2_sq,
            double mom_curr_sq,
            vector< double > r_mom1,
            vector< double > r_n_mom1,
            vector< double > r_mom2,
            vector< double > r2,
            vector< double > r_mom_curr,
            vector< double > r_n_mom_curr )
```

Here is the call graph for this function:



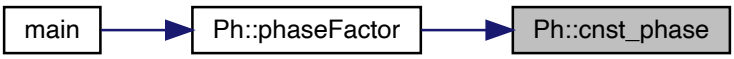
Here is the caller graph for this function:



5.8.2.2 cnst_phase()

```
map< Ph::tripKey, complex< double > > Ph::cnst_phase (
            int twoJ1,
            int twoJ2,
            int twoJCurr )
```

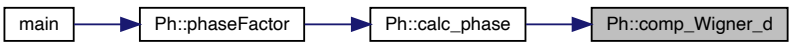
Here is the caller graph for this function:



5.8.2.3 comp_Wigner_d()

```
std::complex< double > Ph::comp_Wigner_d (
            int twoJ,
            int twolam1,
            int twolam2,
            double a1,
            double b1,
            double c1,
            double a2,
            double b2,
            double c2,
            int n )
```

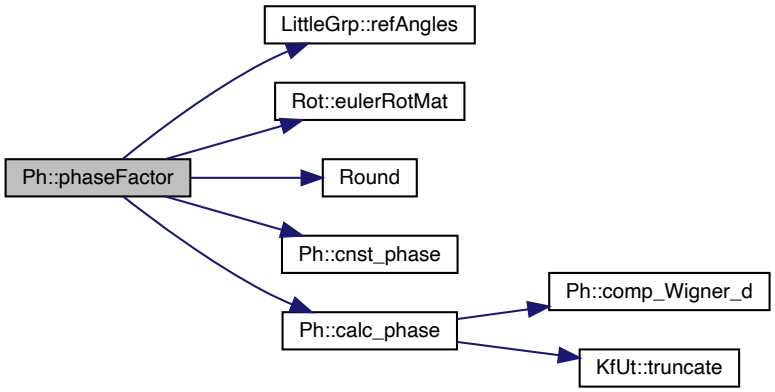
Here is the caller graph for this function:



5.8.2.4 phaseFactor()

```
Ph::phChars Ph::phaseFactor (
            int twoJ1,
            int twoJ2,
            int twoJCurr,
            Eigen::Vector3d mom1,
            Eigen::Vector3d mom2,
            bool compute )
```

Here is the call graph for this function:



Here is the caller graph for this function:

5.9 PolVec Namespace Reference

Functions

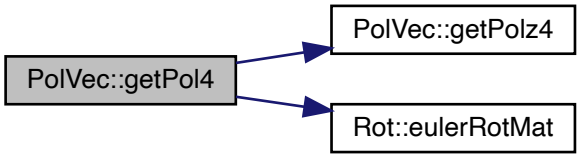
- Eigen::MatrixXcd getPolz4 (double &mom_sq, const int &two_helicity, double &mass_sq, bool &curr)
- Eigen::MatrixXcd getPol4 (double &mom_sq, const int &two_helicity, double &mass_sq, double &phi, double &theta, double &psi, bool curr)

5.9.1 Function Documentation

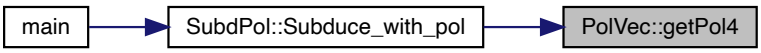
5.9.1.1 getPol4()

```
Eigen::MatrixXcd PolVec::getPol4 (
            double & mom_sq,
            const int & two_helicity,
            double & mass_sq,
            double & phi,
            double & theta,
            double & psi,
            bool curr )
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.1.2 getPolz4()

```
Eigen::MatrixXcd PolVec::getPolz4 (
            double & mom_sq,
            const int & two_helicity,
            double & mass_sq,
            bool & curr )
```

Here is the caller graph for this function:

5.10 Rot Namespace Reference

Functions

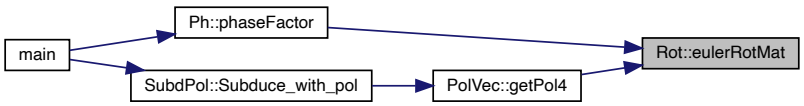
- Eigen::MatrixXd eulerRotMat (double alpha, double beta, double gamma)

5.10.1 Function Documentation

5.10.1.1 eulerRotMat()

```
Eigen::MatrixXd Rot::eulerRotMat (
            double alpha,
            double beta,
            double gamma )
```

Here is the caller graph for this function:



5.11 Subd Namespace Reference

Functions

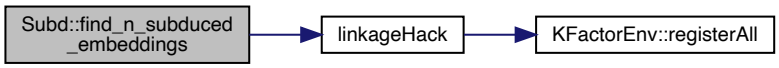
- map< int, complex< double > > subduce_lg_boson (const irrep_label &irrep, const string &little_group)
- map< int, complex< double > > subduce_lg_fermion (const irrep_label &irrep, const string &little_group)
- map< int, complex< double > > subduce_oct (const irrep_label &irrep)
- int find_n_subduced_embeddings (const string &group, const string &irrep, int twoJ, int eta_tilde)

5.11.1 Function Documentation

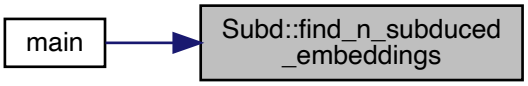
5.11.1.1 find_n_subduced_embeddings()

```
int Subd::find_n_subduced_embeddings (
            const string & group,
            const string & irrep,
            int twoJ,
            int eta_tilde )
```

Here is the call graph for this function:



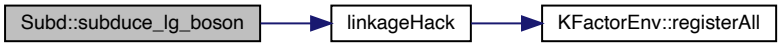
Here is the caller graph for this function:



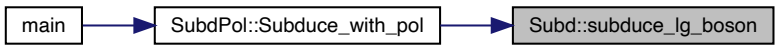
5.11.1.2 subduce_lg_boson()

```
map< int, complex< double > > Subd::subduce_lg_boson (
            const irrep_label & irrep,
            const string & little_group )
```

Here is the call graph for this function:



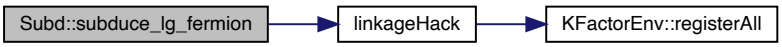
Here is the caller graph for this function:



5.11.1.3 subduce_lg_fermion()

```
map< int, complex< double > > Subd::subduce_lg_fermion (
            const irrep_label & irrep,
            const string & little_group )
```

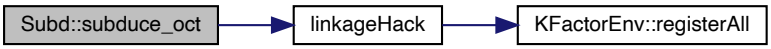
Here is the call graph for this function:



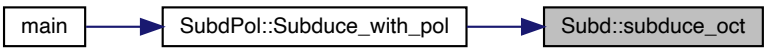
5.11.1.4 subduce_oct()

```
map< int, complex< double > > Subd::subduce_oct (
            const irrep_label & irrep )
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.12 SubdPol Namespace Reference

Functions

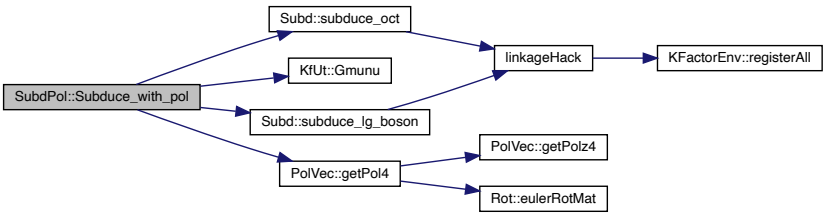
- map< int, Eigen::MatrixXcd > Subduce_with_pol (double &mom_sq, double &mass_sq, int &twoJ, const irrep_label &irrep, const string &little_group, double R1_phi, double R1_theta, double R1_psi, bool curr)

5.12.1 Function Documentation

5.12.1.1 Subduce_with_pol()

```
map< int, Eigen::MatrixXcd > SubdPol::Subduce_with_pol (
            double & mom_sq,
            double & mass_sq,
            int & twoJ,
            const irrep_label & irrep,
            const string & little_group,
            double R1_phi,
            double R1_theta,
            double R1_psi,
            bool curr )
```

Here is the call graph for this function:



Here is the caller graph for this function:

Chapter 6

Data Type Documentation

6.1 hadron Struct Reference

`#include <gen_redstar_xml.h>`

Data Fields

- string name
- Array1dO< string > levels
- int twoJ
- int P
- int ell
- double max_mom
- ADAT::Array1dO< string > elab

6.1.1 Field Documentation

6.1.1.1 elab

`ADAT::Array1dO<string> hadron::elab`

6.1.1.2 ell

`int hadron::ell`

6.1.1.3 levels

```
Array1dO<string> hadron::levels
```

6.1.1.4 max_mom

```
double hadron::max_mom
```

6.1.1.5 name

```
string hadron::name
```

6.1.1.6 P

```
int hadron::P
```

6.1.1.7 twoJ

```
int hadron::twoJ
```

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

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h

6.2 irrep_label Struct Reference

```
#include <subduction.h>
```

Public Member Functions

- bool operator< (const irrep_label &rhs) const

Data Fields

- string irrep
- int row
- int twoJ
- int n
- int P

6.2.1 Member Function Documentation

6.2.1.1 operator<()

```
bool irrep_label::operator< (
            const irrep_label & rhs ) const
```

6.2.2 Field Documentation

6.2.2.1 irrep

```
string irrep_label::irrep
```

6.2.2.2 n

```
int irrep_label::n
```

6.2.2.3 P

```
int irrep_label::P
```

6.2.2.4 row

```
int irrep_label::row
```

6.2.2.5 twoJ

```
int irrep_label::twoJ
```

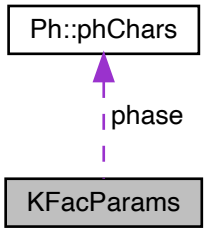
The documentation for this struct was generated from the following file:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h

6.3 KFacParams Class Reference

`#include <kfac_params.h>`

Collaboration diagram for KFacParams:



Public Member Functions

- virtual ~KFacParams ()
- KFacParams (map< int, Eigen::MatrixXcd >, map< int, Eigen::MatrixXcd >, map< int, Eigen::MatrixXcd >, Ph::phChars, VectorXd, VectorXd)
- virtual vector< MatrixXcd > subPhSum () const
- virtual Ph::tripKey two_abs_lam () const

Data Fields

- map< int, Eigen::MatrixXcd > Sub1
- map< int, Eigen::MatrixXcd > SubCurr
- map< int, Eigen::MatrixXcd > Sub3
- Ph::phChars phase
- VectorXd qp
- VectorXd qm

6.3.1 Constructor & Destructor Documentation

6.3.1.1 ~KFacParams()

`virtual KFacParams::~KFacParams ( )  [inline], [virtual]`

6.3.1.2 KFacParams()

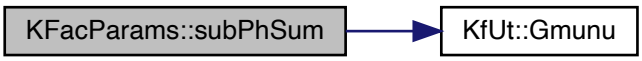
```
KFacParams::KFacParams (
            map< int, Eigen::MatrixXcd > Sub1_,
            map< int, Eigen::MatrixXcd > SubCurr_,
            map< int, Eigen::MatrixXcd > Sub3_,
            Ph::phChars phase_,
            VectorXd qp_,
            VectorXd qm_ )
```

6.3.2 Member Function Documentation

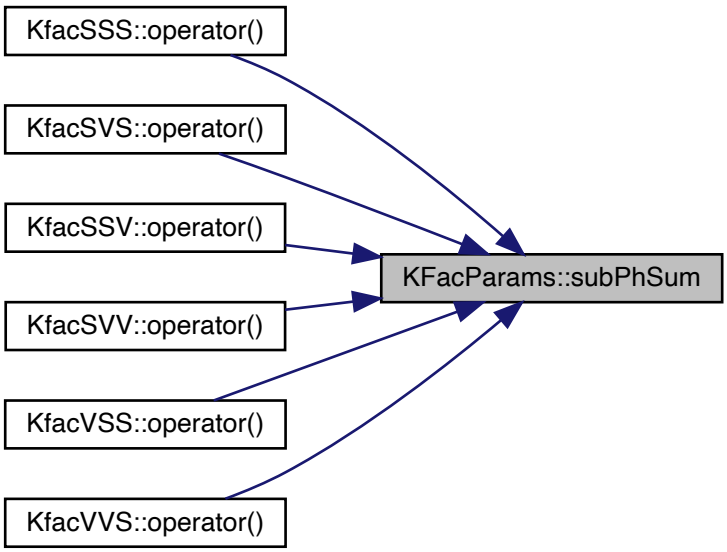
6.3.2.1 subPhSum()

```
vector< MatrixXcd > KFacParams::subPhSum ( ) const  [virtual]
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.2.2 two_abs_lam()

```
Ph::tripKey KFacParams::two_abs_lam ( ) const  [virtual]
```

6.3.3 Field Documentation

6.3.3.1 phase

```
Ph::phChars KFacParams::phase
```

6.3.3.2 qm

```
VectorXd KFacParams::qm
```

6.3.3.3 qp

```
VectorXd KFacParams::qp
```

6.3.3.4 Sub1

```
map< int, Eigen::MatrixXcd > KFacParams::Sub1
```

6.3.3.5 Sub3

```
map< int, Eigen::MatrixXcd > KFacParams::Sub3
```

6.3.3.6 SubCurr

```
map< int, Eigen::MatrixXcd > KFacParams::SubCurr
```

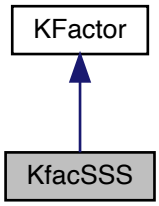
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc

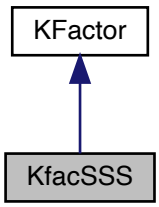
6.4 KfacSSS Class Reference

```
#include <kfactor_pigammapi.h>
```

Inheritance diagram for KfacSSS:



Collaboration diagram for KfacSSS:



Public Member Functions

- vector< complex< double > > operator() (const KFacParams &params) const
- string name () const

6.4.1 Member Function Documentation

6.4.1.1 name()

```
string KfacSSS::name ( ) const  [inline], [virtual]
```

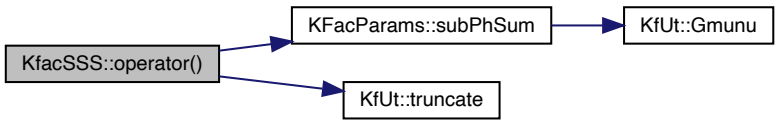
Implements KFactor.

6.4.1.2 operator()()

```
vector< complex< double > > KfacSSS::operator() (
            const KFacParams & params ) const  [virtual]
```

Implements KFactor.

Here is the call graph for this function:



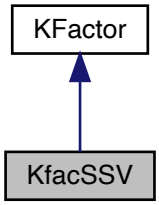
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc

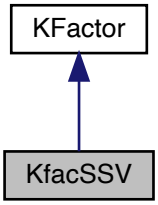
6.5 KfacSSV Class Reference

```
#include <kfactor_pigammarho.h>
```

Inheritance diagram for KfacSSV:



Collaboration diagram for KfacSSV:



Public Member Functions

- vector< complex< double > > operator() (const KFacParams &params) const
- string name () const

6.5.1 Member Function Documentation

6.5.1.1 name()

```
string KfacSSV::name ( ) const  [inline], [virtual]
```

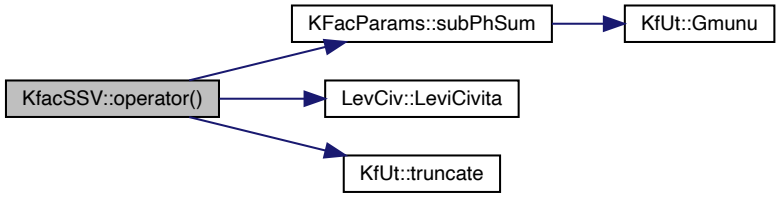
Implements KFactor.

6.5.1.2 operator()()

```
vector< complex< double > > KfacSSV::operator() (
            const KFacParams & params ) const  [virtual]
```

Implements KFactor.

Here is the call graph for this function:



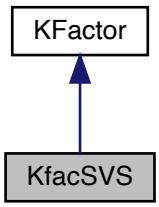
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc

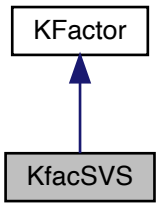
6.6 KfacSVS Class Reference

`#include <kfactor_pigammapi.h>`

Inheritance diagram for KfacSVS:



Collaboration diagram for KfacSVS:



Public Member Functions

- vector< complex< double > > operator() (const KFacParams &params) const
- string name () const

6.6.1 Member Function Documentation

6.6.1.1 name()

```
string KfacSVS::name ( ) const  [inline], [virtual]
```

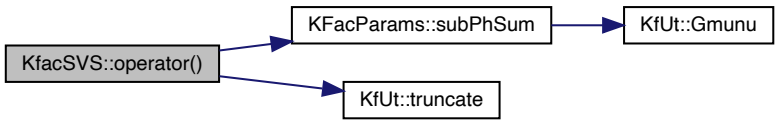
Implements KFactor.

6.6.1.2 operator()()

```
vector< complex< double > > KfacSVS::operator() (
            const KFacParams & params ) const  [virtual]
```

Implements KFactor.

Here is the call graph for this function:



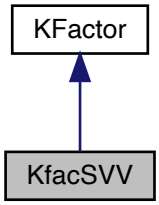
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc

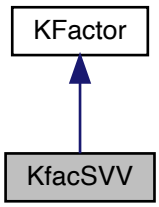
6.7 KfacSVV Class Reference

```
#include <kfactor_pigammarho.h>
```

Inheritance diagram for KfacSVV:



Collaboration diagram for KfacSVV:



Public Member Functions

- vector< complex< double > > operator() (const KFacParams &params) const
- string name () const

6.7.1 Member Function Documentation

6.7.1.1 name()

```
string KfacSVV::name ( ) const  [inline], [virtual]
```

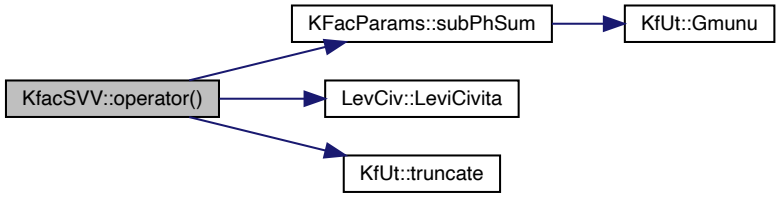
Implements KFactor.

6.7.1.2 operator()()

```
vector< complex< double > > KfacSVV::operator() (
            const KFacParams & params ) const  [virtual]
```

Implements KFactor.

Here is the call graph for this function:



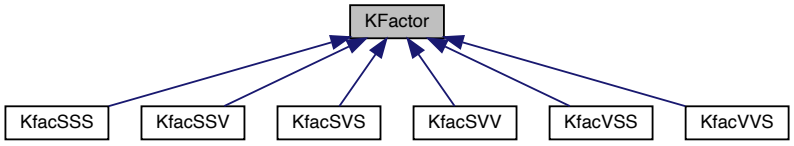
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc

6.8 KFactor Class Reference

```
#include <kfactor_pigammarho.h>
```

Inheritance diagram for KFactor:



Public Member Functions

- virtual ~KFactor ()
- virtual vector< complex< double > > operator() (const KFacParams &params) const =0
- virtual string name () const =0

6.8.1 Constructor & Destructor Documentation

6.8.1.1 ~KFactor()

```
virtual KFactor::~KFactor ( )  [inline], [virtual]
```

6.8.2 Member Function Documentation

6.8.2.1 name()

```
virtual string KFactor::name ( ) const  [pure virtual]
```

Implemented in KfacVSS, KfacVVS, KfacSSV, KfacSVV, KfacSSS, and KfacSVS.

6.8.2.2 operator()()

```
virtual vector<complex<double> > KFactor::operator() (
            const KFacParams & params ) const  [pure virtual]
```

Implemented in KfacVSS, KfacVVS, KfacSSV, KfacSVV, KfacSSS, and KfacSVS.

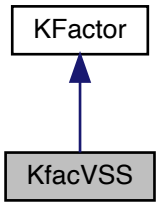
The documentation for this class was generated from the following file:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h

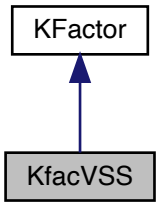
6.9 KfacVSS Class Reference

```
#include <kfactor_pigammarho.h>
```

Inheritance diagram for KfacVSS:



Collaboration diagram for KfacVSS:



Public Member Functions

- vector< complex< double > > operator() (const KFacParams &params) const
- string name () const

6.9.1 Member Function Documentation

6.9.1.1 name()

```
string KfacVSS::name ( ) const  [inline], [virtual]
```

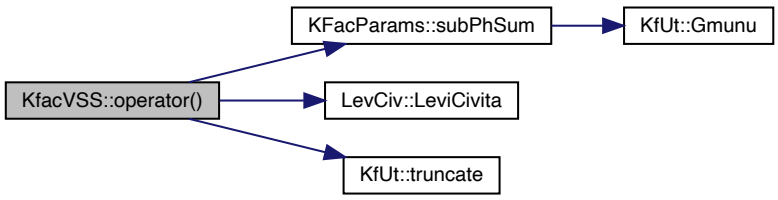
Implements KFactor.

6.9.1.2 operator()()

```
vector< complex< double > > KfacVSS::operator() (
            const KFacParams & params ) const  [virtual]
```

Implements KFactor.

Here is the call graph for this function:



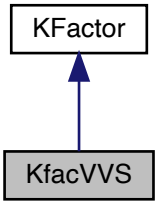
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc

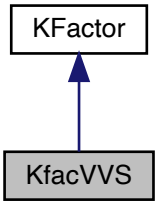
6.10 KfacVVS Class Reference

`#include <kfactor_pigammarho.h>`

Inheritance diagram for KfacVVS:



Collaboration diagram for KfacVVS:



Public Member Functions

- vector< complex< double > > operator() (const KFacParams &params) const
- string name () const

6.10.1 Member Function Documentation

6.10.1.1 name()

`string KfacVVS::name ( ) const [inline], [virtual]`

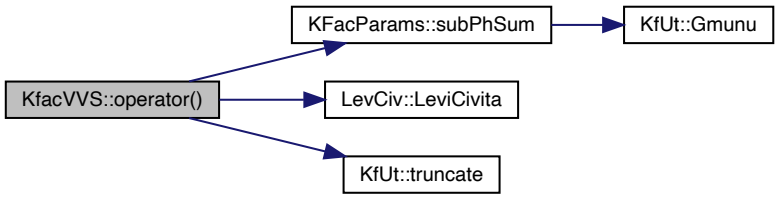
Implements KFactor.

6.10.1.2 operator()()

```
vector< complex< double > > KfacVVS::operator() (
            const KFacParams & params ) const  [virtual]
```

Implements KFactor.

Here is the call graph for this function:



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

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc

6.11 Ph::phChars Struct Reference

```
#include <phase.h>
```

Public Member Functions

- bool operator< (const phChars &rhs) const

Data Fields

- Eigen::Vector3d mom2
- Eigen::Vector3d mom1
- map< Ph::tripKey, complex< double > > lam_phase
- Eigen::MatrixXcd r

6.11.1 Member Function Documentation

6.11.1.1 operator<()

```
bool Ph::phChars::operator< (
            const phChars & rhs ) const
```

6.11.2 Field Documentation

6.11.2.1 lam_phase

```
map< Ph::tripKey , complex<double> > Ph::phChars::lam_phase
```

6.11.2.2 mom1

```
Eigen::Vector3d Ph::phChars::mom1
```

6.11.2.3 mom2

```
Eigen::Vector3d Ph::phChars::mom2
```

6.11.2.4 r

```
Eigen::MatrixXcd Ph::phChars::r
```

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

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h

6.12 KfUt::ToArray Class Reference

```
#include <kfac_utils.h>
```

Static Public Member Functions

- static XMLArray::Array< int > toArray (Eigen::Vector3d input)
- static XMLArray::Array< int > toArray (Array1dO< int > input)

6.12.1 Member Function Documentation

6.12.1.1 toArray() [1/2]

```
XMLArray::Array< int > KfUt::ToArray::toArray (
            Eigen::Vector3d input )  [static]
```

Here is the caller graph for this function:

6.12.1.2 toArray() [2/2]

```
XMLArray::Array< int > KfUt::ToArray::toArray (
            Array1dO< int > input )  [static]
```

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

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc

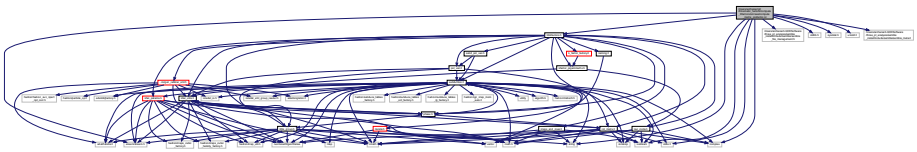
Chapter 7

File Documentation

7.1 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_↩ prefactor.cc File Reference

```
#include "lib/kfactors.h"
#include "/Users/archanar/LQCDSoftware/three_pt_analysis/semble_install/include/semble/s
_file_management.h"
#include "/Users/archanar/LQCDSoftware/three_pt_analysis/semble_install/include/semble/s
_meta.h"
```

Include dependency graph for compute_matrix_prefactor.cc:



Functions

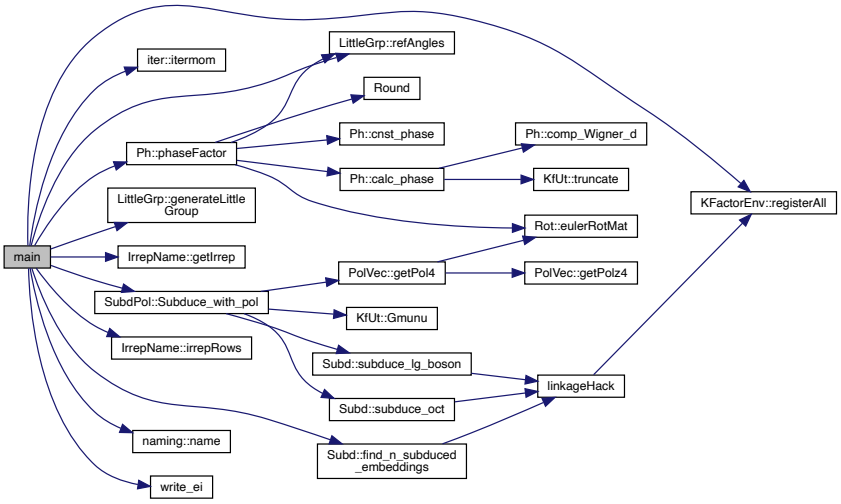
- int main (int argc, char **argv)

7.1.1 Function Documentation

7.1.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

Here is the call graph for this function:



7.2 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.cc File Reference

```
#include "gen_redstar_xml.h"
```

Functions

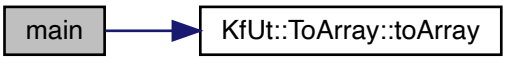
- int main (int argc, char **argv)

7.2.1 Function Documentation

7.2.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

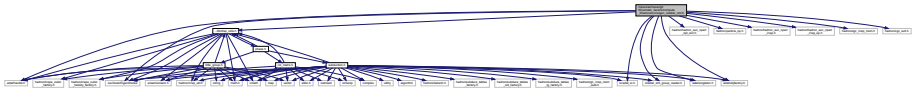
Here is the call graph for this function:



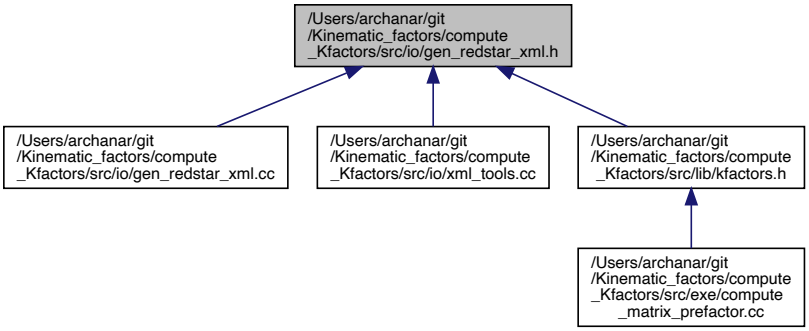
7.3 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h File Reference

```
#include </usr/local/Eigen/Dense>
#include "io/adat_io.h"
#include "io/adat_xml_group_reader.h"
#include "hadron/hadron_sun_npart_npt_corr.h"
#include "hadron/particle_op.h"
#include "hadron/hadron_sun_npart_irrep.h"
#include "hadron/hadron_sun_npart_irrep_op.h"
#include "adat/singleton.h"
#include "adat/objfactory.h"
#include <adat/handle.h>
#include "hadron/cgc_irrep_mom.h"
#include "hadron/cgc_su3.h"
#include "../lib/kfac_utils.h"
```

Include dependency graph for gen_redstar_xml.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct hadron

Functions

- void write_ei (XMLWriter &xml, const std::string &path, const Eigen::Vector3d &input)

7.3.1 Function Documentation

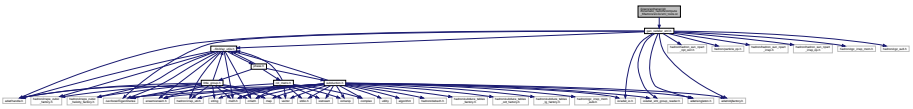
7.3.1.1 write_ei()

```
void write_ei (
            XMLWriter & xml,
            const std::string & path,
            const Eigen::Vector3d & input )
```

7.4 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/xml_tools.cc File Reference

```
#include "gen_redstar_xml.h"
```

Include dependency graph for xml_tools.cc:



Functions

- void write_ei (XMLWriter &xml, const std::string &path, const Eigen::Vector3d &input)

7.4.1 Function Documentation

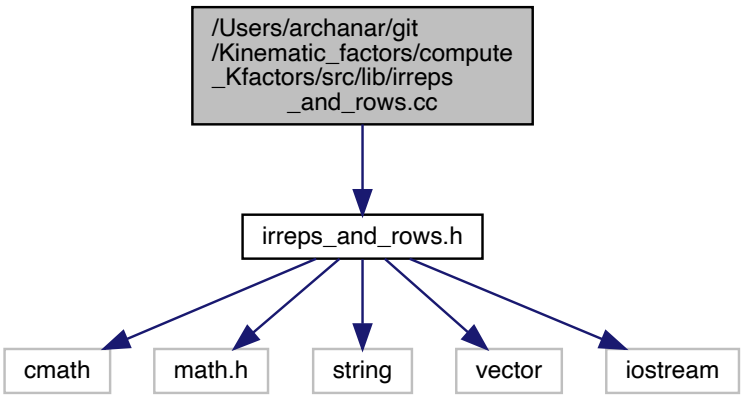
7.4.1.1 write_ei()

```
void write_ei (
            XMLWriter & xml,
            const std::string & path,
            const Eigen::Vector3d & input )
```

7.5 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.cc File Reference

```
#include "irreps_and_rows.h"
```

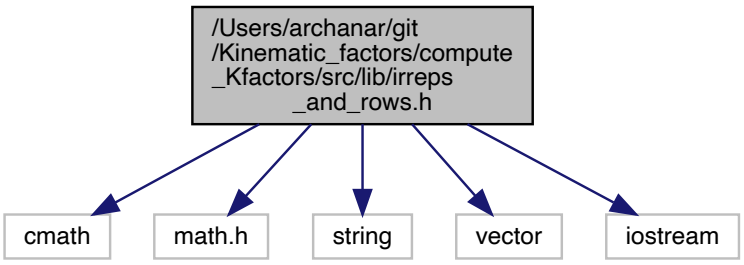
Include dependency graph for irreps_and_rows.cc:



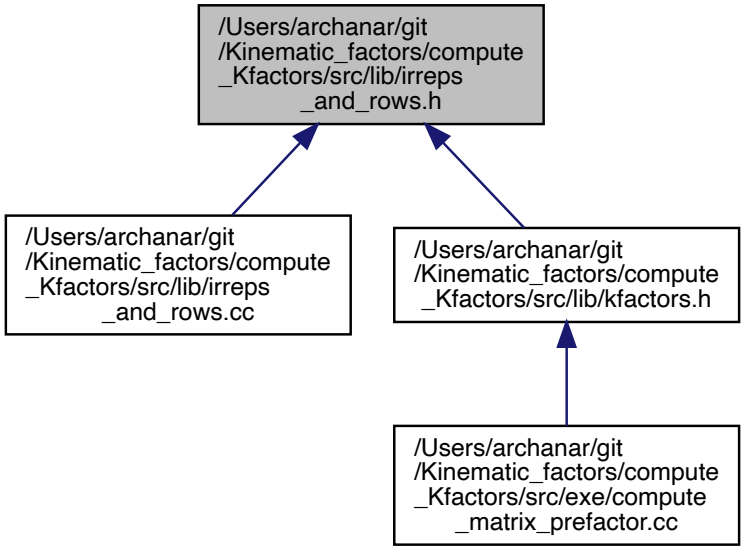
7.6 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h File Reference

```
#include <cmath>
#include "math.h"
#include <string>
#include <vector>
#include <iostream>
```

Include dependency graph for irreps_and_rows.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- IrrepName

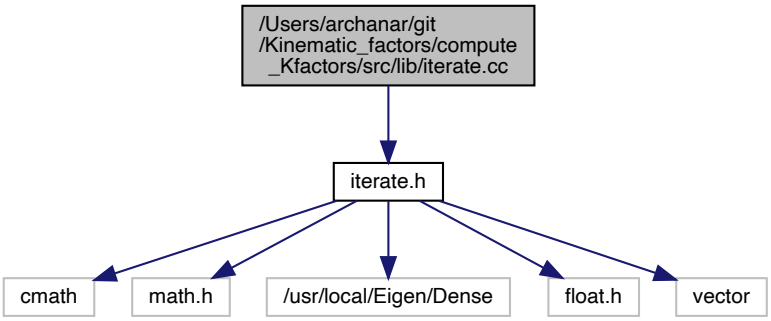
Functions

- std::vector< std::string > IrrepName::getIrrep (int &twoJ, int &P, string &lg)
- int IrrepName::irrepRows (string &irrep)

7.7 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.cc File Reference

```
#include "iterate.h"
```

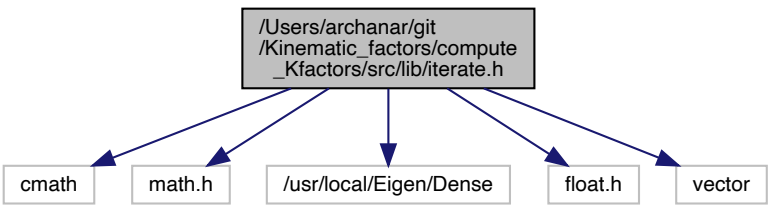
Include dependency graph for iterate.cc:



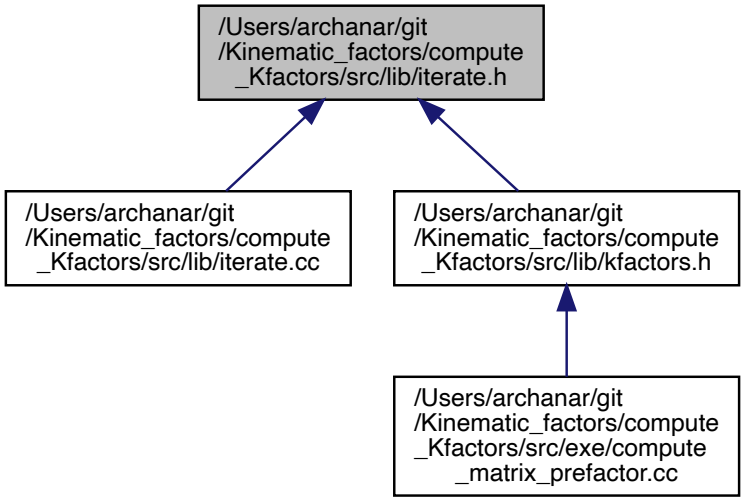
7.8 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h File Reference

```
#include <cmath>
#include "math.h"
#include </usr/local/Eigen/Dense>
#include <float.h>
#include <vector>
```

Include dependency graph for iterate.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- iter

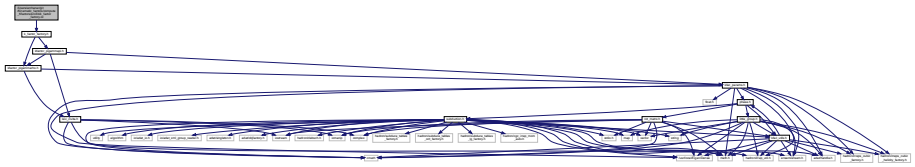
Functions

- std::vector< Vector3d > iter::itermom (double max_mom)

7.9 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.cc File Reference

```
#include "k_factor_factory.h"
```

Include dependency graph for k_factor_factory.cc:



Namespaces

- KFactorEnv

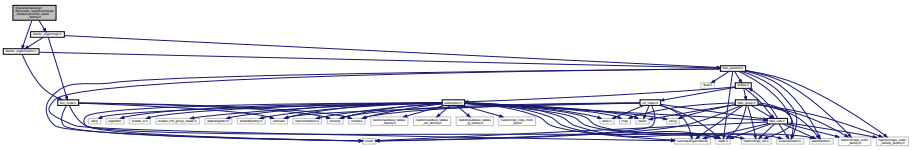
Functions

- bool KFactorEnv::registerAll ()

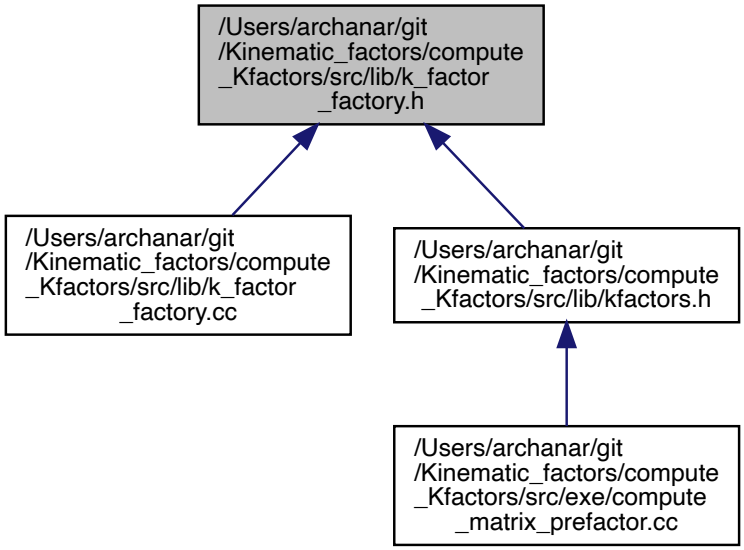
7.10 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.h File Reference

```
#include "kfactor_pigammarho.h"
#include "kfactor_pigammapi.h"
```

Include dependency graph for k_factor_factory.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- KFactorEnv

Typedefs

- typedef SingletonHolder< ObjectFactory< KFactor, string, TYPELIST_2(XMLReader &, const string &), KFactor *(*)(XMLReader &, const string &), StringFactoryError > > TheKFactorFactory

Functions

- bool KFactorEnv::registerAll ()

7.10.1 Typedef Documentation

7.10.1.1 TheKFactorFactory

```
typedef SingletonHolder< ObjectFactory<KFactor, string, TYPELIST_2( XMLReader&, const string&),
KFactor* (*)(XMLReader&, const string&), StringFactoryError> > TheKFactorFactory
```

7.11 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc File Reference

```
#include "kfac_params.h"
```

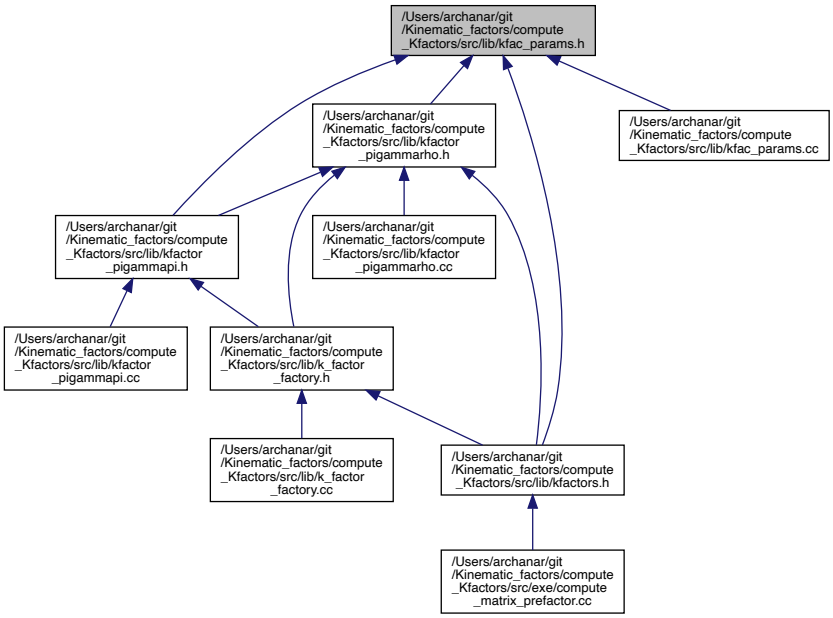
Include dependency graph for kfac_params.cc:

7.12 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h File Reference

```
#include <cmath>
#include "math.h"
#include </usr/local/Eigen/Dense>
#include <float.h>
#include "phase.h"
#include <adat/handle.h>
#include "hadron/irreps_cubic_factory.h"
#include "hadron/irreps_cubic_helicity_factory.h"
#include "hadron/irrep_util.h"
#include "ensem/ensem.h"
```

Include dependency graph for kfac_params.h:

This graph shows which files directly or indirectly include this file:



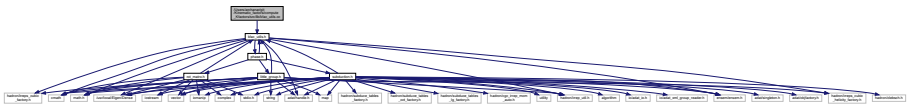
Data Structures

- class KFacParams

7.13 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc File Reference

```
#include "kfac_utils.h"
```

Include dependency graph for kfac_utils.cc:



Namespaces

- KfUt

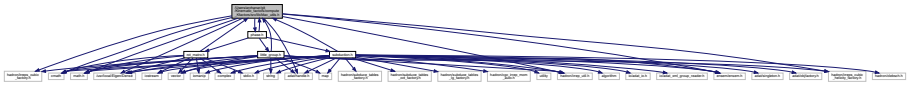
Functions

- double KfUt::truncate (double num, int precision)
- Eigen::MatrixXcd KfUt::Gmunu ()

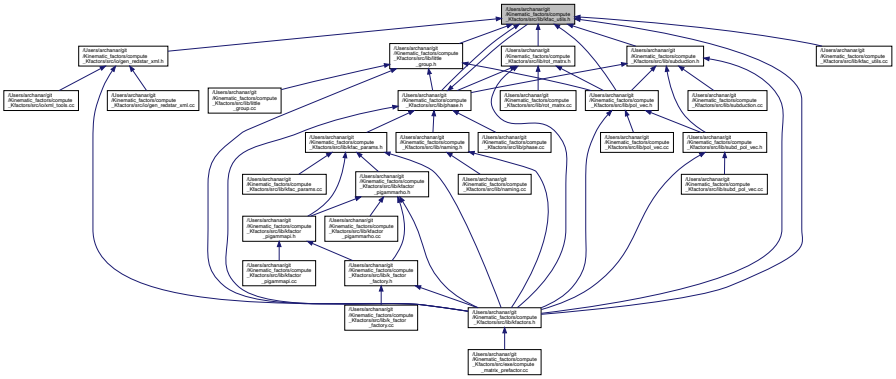
7.14 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h File Reference

```
#include <cmath>
#include "math.h"
#include </usr/local/Eigen/Dense>
#include "phase.h"
#include <adat/handle.h>
#include "hadron/irreps_cubic_factory.h"
#include "hadron/irreps_cubic_helicity_factory.h"
#include "hadron/irrep_util.h"
#include "ensem/ensem.h"
```

Include dependency graph for kfac_utils.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KfUt::ToArray

Namespaces

- KfUt

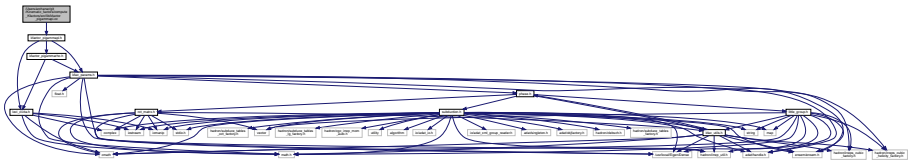
Functions

- double KfUt::truncate (double num, int precision)
- Eigen::MatrixXcd KfUt::Gmunu ()

7.15 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc File Reference

```
#include "kfactor_pigammapi.h"
```

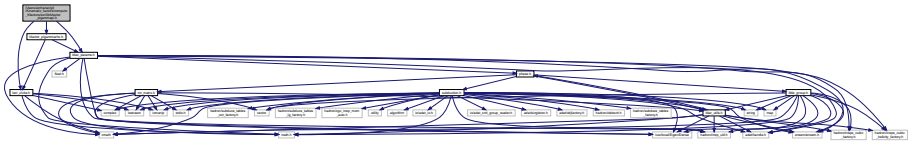
Include dependency graph for kfactor_pigammapi.cc:



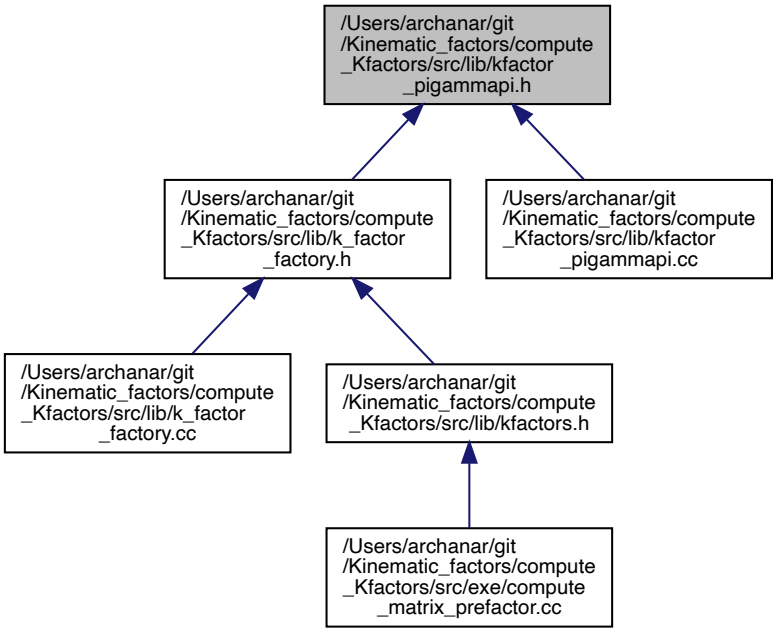
7.16 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h File Reference

```
#include "kfac_params.h"
#include "levi_civita.h"
#include "kfactor_pigammarho.h"
```

Include dependency graph for kfactor_pigammapi.h:



This graph shows which files directly or indirectly include this file:



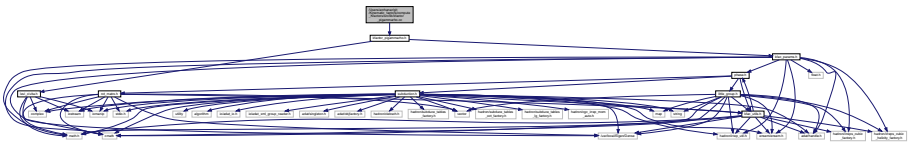
Data Structures

- class KfacSVS
- class KfacSSS

7.17 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc File Reference

```
#include "kfactor_pigammarho.h"
```

Include dependency graph for kfactor_pigammarho.cc:

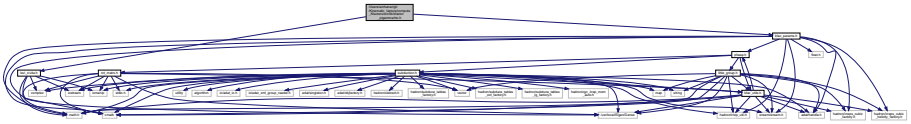


7.18 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h File Reference

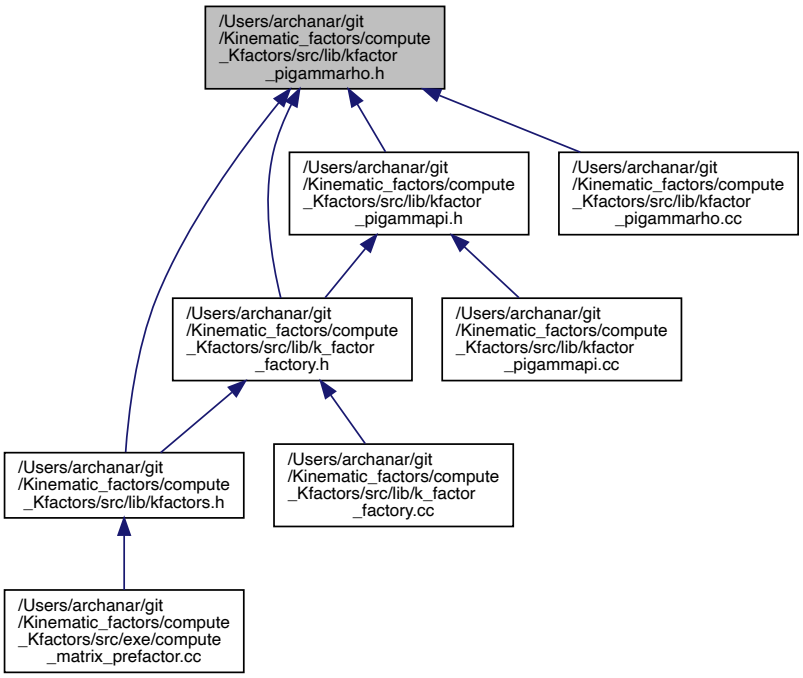
```
#include "kfac_params.h"
```

```
#include "levi_civita.h"
```

Include dependency graph for kfactor_pigammarho.h:



This graph shows which files directly or indirectly include this file:



Data Structures

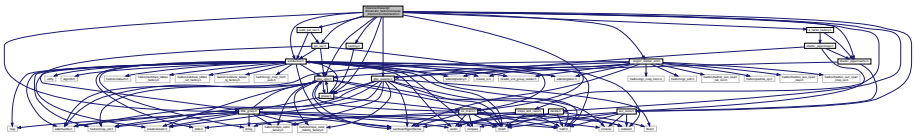
- class KFactor
- class KfacSVV
- class KfacSSV
- class KfacVVS
- class KfacVSS

7.19 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h File Reference

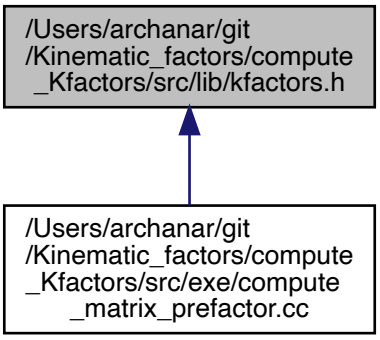
```
#include "subduction.h"
#include "subd_pol_vec.h"
#include "pol_vec.h"
```

```
#include "little_group.h"
#include "levi_civita.h"
#include "kfac_utils.h"
#include "rot_matrx.h"
#include "irreps_and_rows.h"
#include "phase.h"
#include "../io/gen_redstar_xml.h"
#include "k_factor_factory.h"
#include "kfac_params.h"
#include "kfactor_pigammarho.h"
#include "naming.h"
#include "iterate.h"
```

Include dependency graph for kfactors.h:



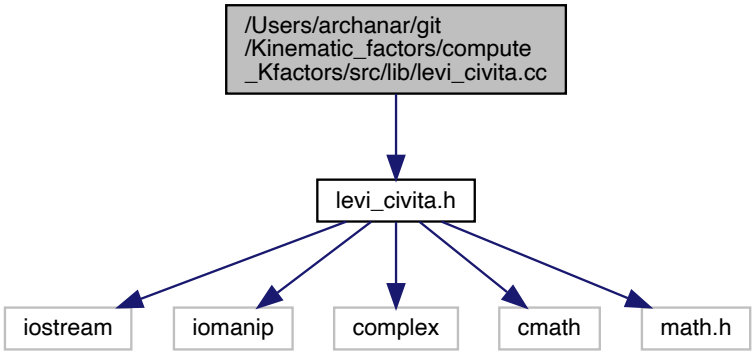
This graph shows which files directly or indirectly include this file:



7.20 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.cc File Reference

```
#include "levi_civita.h"
```

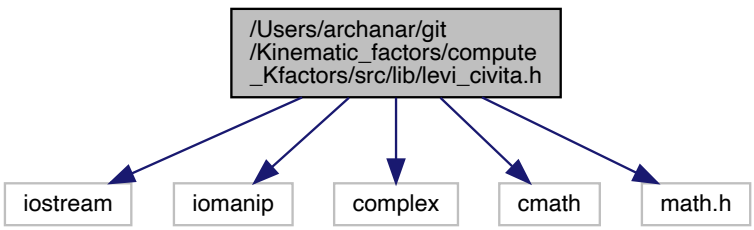
Include dependency graph for levi_civita.cc:



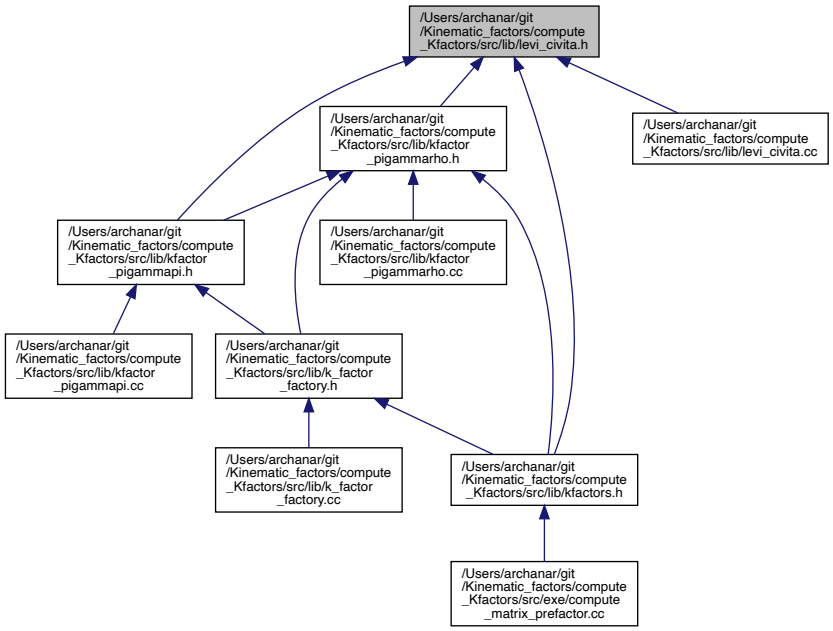
7.21 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.h File Reference

```
#include <iostream>
#include <iomanip>
#include <complex>
#include <cmath>
#include "math.h"
```

Include dependency graph for levi_civita.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- LevCiv

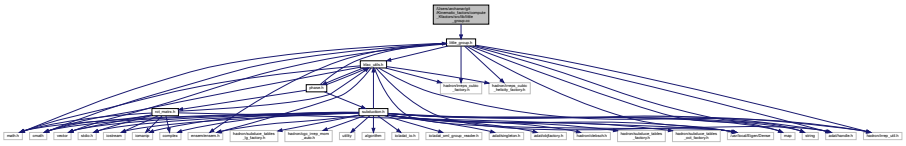
Functions

- double LevCiv::LeviCivita (int arr[ ], int n)

7.22 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.cc File Reference

```
#include "little_group.h"
```

Include dependency graph for little_group.cc:



Namespaces

- LittleGrp

Functions

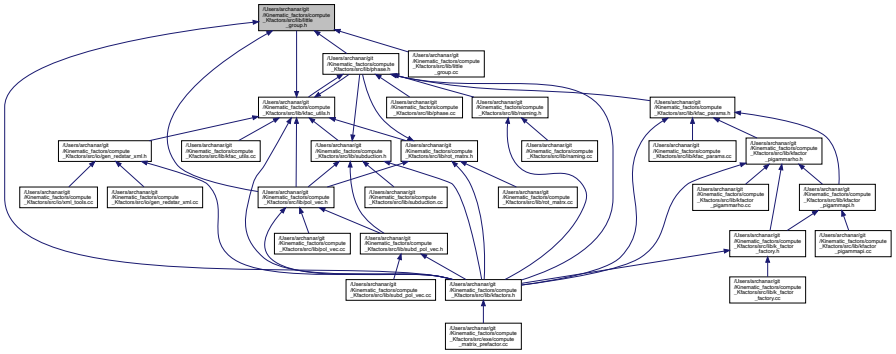
- string LittleGrp::generateLittleGroup (Eigen::Vector3d &mom_)
- std::vector< double > LittleGrp::refAngles (Eigen::Vector3d mom1)

7.23 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.h File Reference

```
#include <vector>
#include <map>
#include <string>
#include <cmath>
#include "math.h"
#include </usr/local/Eigen/Dense>
#include "kfac_utils.h"
#include <adat/handle.h>
#include "hadron/irreps_cubic_factory.h"
#include "hadron/irreps_cubic_helicity_factory.h"
#include "hadron/irrep_util.h"
#include "ensem/ensem.h"
```

Include dependency graph for little_group.h:

This graph shows which files directly or indirectly include this file:



Namespaces

- LittleGrp

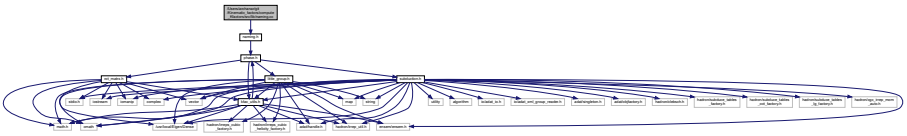
Functions

- string LittleGrp::generateLittleGroup (Eigen::Vector3d &mom_)
- std::vector< double > LittleGrp::refAngles (Eigen::Vector3d mom1)

7.24 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc File Reference

```
#include "naming.h"
```

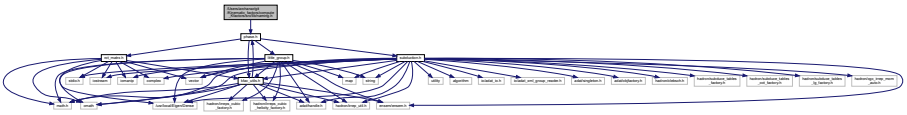
Include dependency graph for naming.cc:



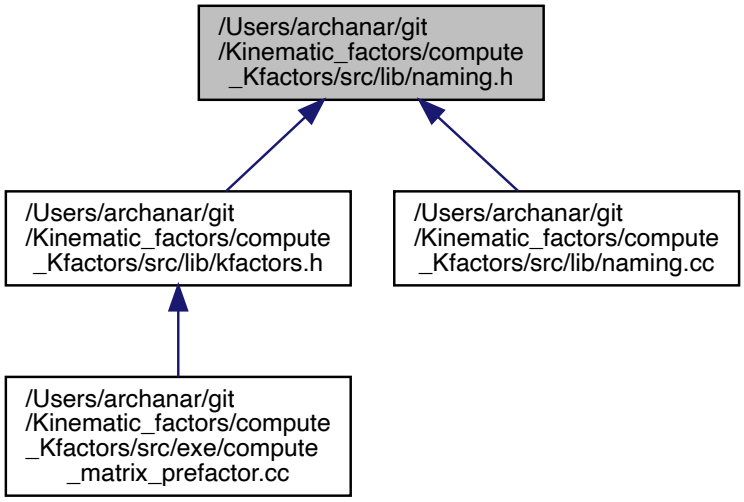
7.25 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h File Reference

```
#include "phase.h"
```

Include dependency graph for naming.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- naming

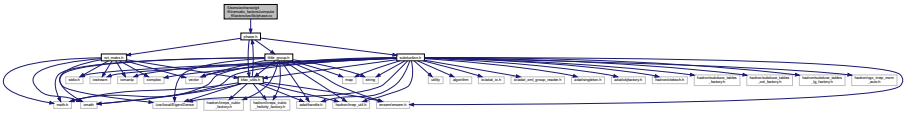
Functions

- string naming::name (int npt, Ph::tripKey two_abs_lam, Vector3d mom1, Vector3d mom_curr, Vector3d mom3, irrep_label rep1, irrep_label rep_curr, irrep_label rep3, string LG1, string LG_curr, string LG3, string lev1, string lev3)

7.26 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.cc File Reference

```
#include "phase.h"
```

Include dependency graph for phase.cc:



Functions

- double Round (double x)

7.26.1 Function Documentation

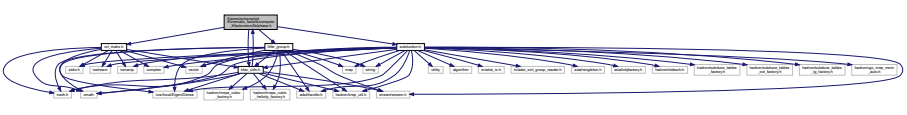
7.26.1.1 Round()

```
double Round (
            double x )
```

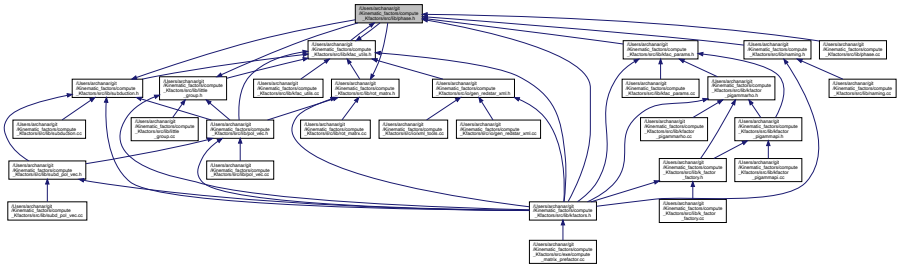
7.27 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h File Reference

```
#include "rot_matrx.h"
#include "little_group.h"
#include "kfac_utils.h"
#include "subduction.h"
```

Include dependency graph for phase.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct Ph::phChars

Namespaces

- Ph

Typedefs

- typedef std::tuple< int, int, int > Ph::tripKey

Functions

- double Round (double x)
- Ph::phChars Ph::phaseFactor (int twoJ1, int twoJ2, int twoJCurr, Eigen::Vector3d mom1, Eigen::Vector3d mom2, bool compute)
- std::complex< double > Ph::comp_Wigner_d (int twoJ, int twolam1, int twolam2, double a1, double b1, double c1, double a2, double b2, double c2, int n)
- map< Ph::tripKey, complex< double > > Ph::calc_phase (int twoJ1, int twoJ2, int twoJCurr, double mom1_sq, double mom2_sq, double mom_curr_sq, vector< double > r_mom1, vector< double > r_n_mom1, vector< double > r_mom2, vector< double > r2, vector< double > r_mom_curr, vector< double > r_n_mom_curr)
- map< Ph::tripKey, complex< double > > Ph::cnst_phase (int twoJ1, int twoJ2, int twoJCurr)

7.27.1 Function Documentation

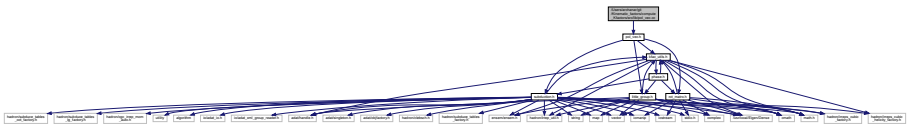
7.27.1.1 Round()

```
double Round (
            double x )
```

7.28 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.cc File Reference

```
#include "pol_vec.h"
```

Include dependency graph for pol_vec.cc:



Namespaces

- PolVec

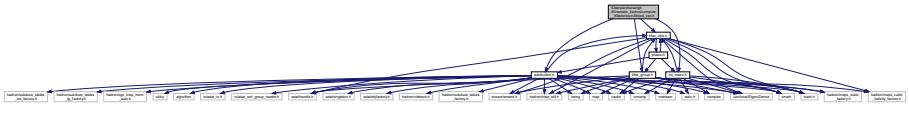
Functions

- Eigen::MatrixXcd PolVec::getPolz4 (double &mom_sq, const int &two_helicity, double &mass_sq, bool &curr)
- Eigen::MatrixXcd PolVec::getPol4 (double &mom_sq, const int &two_helicity, double &mass_sq, double &phi, double &theta, double &psi, bool curr)

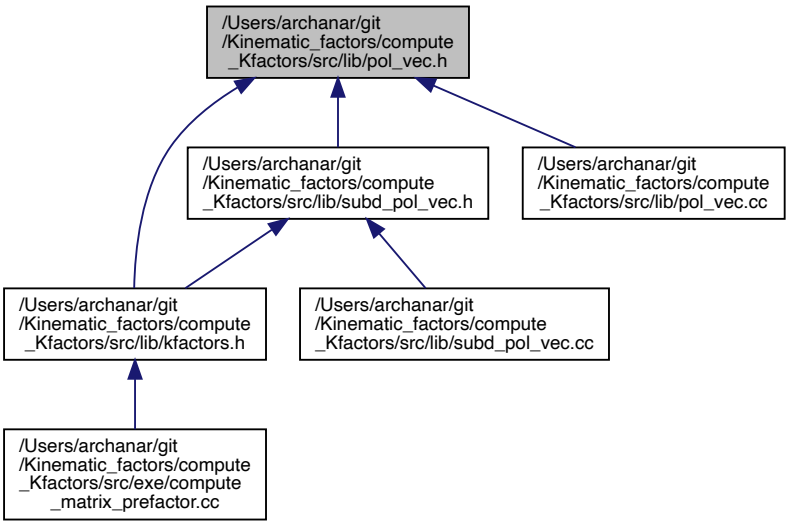
7.29 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.h File Reference

```
#include "kfac_utils.h"
#include "rot_matrx.h"
#include "little_group.h"
#include "subduction.h"
```

Include dependency graph for pol_vec.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- PolVec

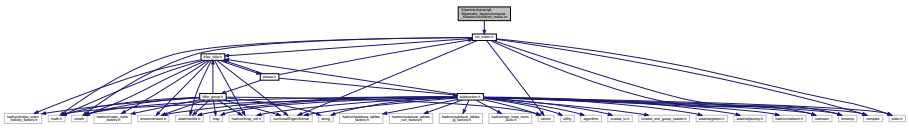
Functions

- Eigen::MatrixXcd PolVec::getPolz4 (double &mom_sq, const int &two_helicity, double &mass_sq, bool &curr)
- Eigen::MatrixXcd PolVec::getPol4 (double &mom_sq, const int &two_helicity, double &mass_sq, double &phi, double &theta, double &psi, bool curr)

7.30 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.cc File Reference

```
#include "rot_matrx.h"
```

Include dependency graph for rot_matrx.cc:



Namespaces

- Rot

Functions

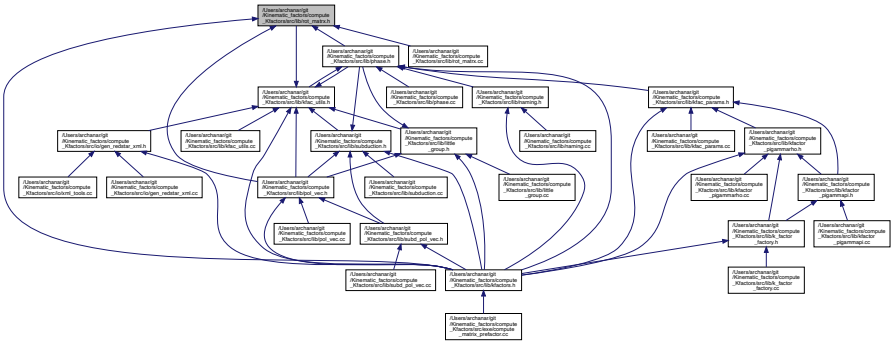
- Eigen::MatrixXd Rot::eulerRotMat (double alpha, double beta, double gamma)

7.31 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.h File Reference

```
#include "kfac_utils.h"
#include <iostream>
#include <iomanip>
#include <complex>
#include <cmath>
#include "math.h"
#include <stdio.h>
#include </usr/local/Eigen/Dense>
#include <vector>
```

Include dependency graph for rot_matrx.h:

This graph shows which files directly or indirectly include this file:



Namespaces

- Rot

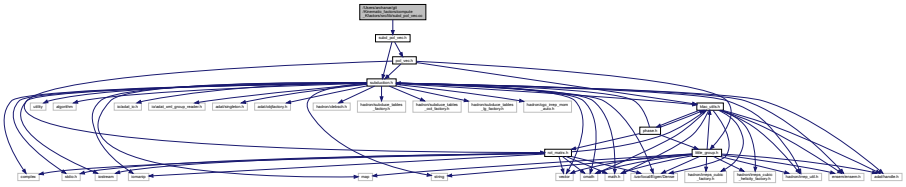
Functions

- Eigen::MatrixXd Rot::eulerRotMat (double alpha, double beta, double gamma)

7.32 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.cc File Reference

```
#include "subd_pol_vec.h"
```

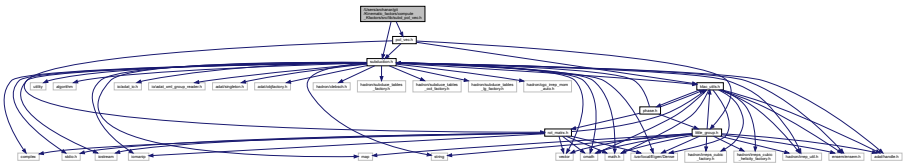
Include dependency graph for subd_pol_vec.cc:



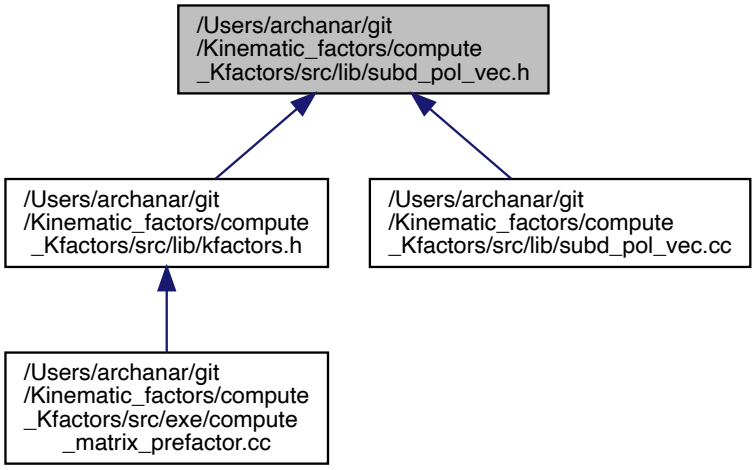
7.33 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.h File Reference

```
#include "subduction.h"
#include "pol_vec.h"
```

Include dependency graph for subd_pol_vec.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- SubdPol

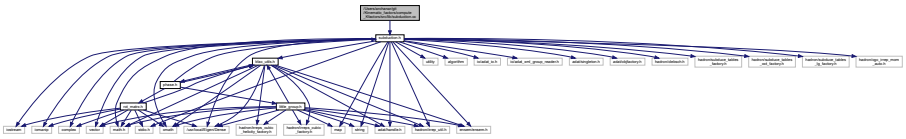
Functions

- map< int, Eigen::MatrixXcd > SubdPol::Subduce_with_pol (double &mom_sq, double &mass_sq, int &twoJ, const irrep_label &irrep, const string &little_group, double R1_phi, double R1_theta, double R1_psi, bool curr)

7.34 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.cc File Reference

```
#include "subduction.h"
```

Include dependency graph for subduction.cc:



Functions

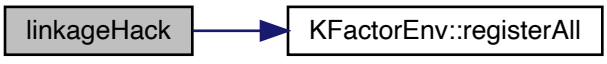
- bool linkageHack (void)

7.34.1 Function Documentation

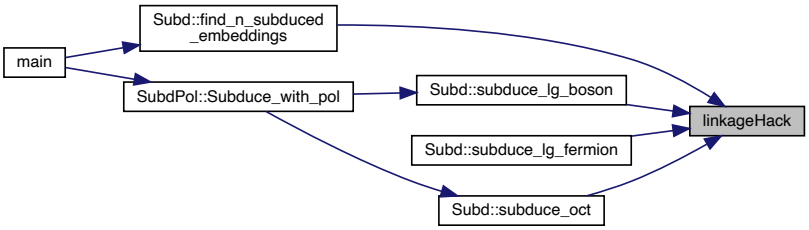
7.34.1.1 linkageHack()

```
bool linkageHack (
            void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:

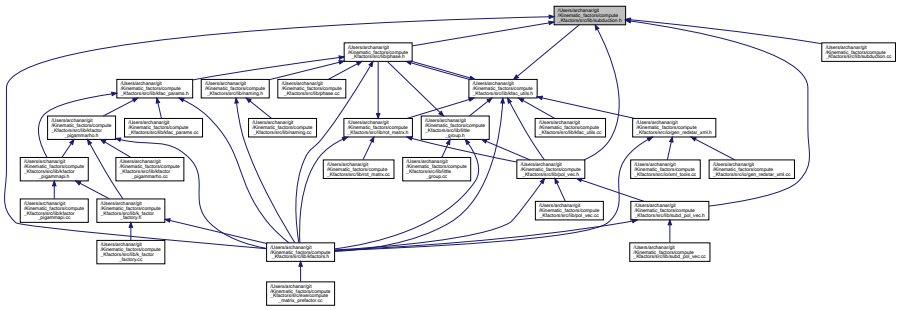


7.35 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h File Reference

```
#include <vector>
#include <iostream>
#include <iomanip>
#include <map>
#include <string>
#include <complex>
#include <utility>
#include <algorithm>
#include <cmath>
#include "math.h"
#include <stdio.h>
#include </usr/local/Eigen/Dense>
#include "io/adat_io.h"
#include "io/adat_xml_group_reader.h"
#include "adat/singleton.h"
#include "adat/objfactory.h"
#include <adat/handle.h>
#include "hadron/clebsch.h"
#include "hadron/subduce_tables_factory.h"
#include "hadron/subduce_tables_oct_factory.h"
#include "hadron/subduce_tables_lg_factory.h"
#include "hadron/cgc_irrep_mom_auto.h"
#include "hadron/irrep_util.h"
#include "ensem/ensem.h"
#include "kfac_utils.h"
```

Include dependency graph for subduction.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- struct irrep_label

Namespaces

- Subd

Functions

- map< int, complex< double > > Subd::subduce_lg_boson (const irrep_label &irrep, const string &little_↩ group)
- map< int, complex< double > > Subd::subduce_lg_fermion (const irrep_label &irrep, const string &little_↩ group)
- map< int, complex< double > > Subd::subduce_oct (const irrep_label &irrep)
- int Subd::find_n_subduced_embeddings (const string &group, const string &irrep, int twoJ, int eta_tilde)

Index