

# Kinematic Factor

1.0

Generated by Doxygen 1.8.15



<b>1 Modules Index</b>	<b>1</b>
1.1 Modules List	1
<b>2 Data Type Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Data Type Index</b>	<b>5</b>
3.1 Data Types List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Module Documentation</b>	<b>9</b>
5.1 IrrepName Namespace Reference	9
5.1.1 Function Documentation	9
5.1.1.1 getIrrep()	9
5.1.1.2 irrepRows()	10
5.2 iter Namespace Reference	10
5.2.1 Function Documentation	10
5.2.1.1 itermom()	10
5.3 KFactorEnv Namespace Reference	10
5.3.1 Function Documentation	11
5.3.1.1 registerAll()	11
5.4 KfUt Namespace Reference	11
5.4.1 Function Documentation	11
5.4.1.1 Gmunu()	12
5.4.1.2 truncate()	12
5.5 LevCiv Namespace Reference	13
5.5.1 Function Documentation	13
5.5.1.1 LeviCivita()	13
5.6 LittleGrp Namespace Reference	13
5.6.1 Function Documentation	13
5.6.1.1 generateLittleGroup()	14
5.6.1.2 refAngles()	14
5.7 naming Namespace Reference	14
5.7.1 Function Documentation	14
5.7.1.1 name()	15
5.8 Ph Namespace Reference	15
5.8.1 Typedef Documentation	16
5.8.1.1 tripKey	16
5.8.2 Function Documentation	16
5.8.2.1 calc_phase()	16
5.8.2.2 cnst_phase()	17
5.8.2.3 comp_Wigner_d()	17

5.8.2.4 phaseFactor()	18
5.9 PolVec Namespace Reference	18
5.9.1 Function Documentation	18
5.9.1.1 getPol4()	19
5.9.1.2 getPolz4()	19
5.10 Rot Namespace Reference	20
5.10.1 Function Documentation	20
5.10.1.1 eulerRotMat()	20
5.11 Subd Namespace Reference	20
5.11.1 Function Documentation	20
5.11.1.1 find_n_subduced_embeddings()	21
5.11.1.2 subduce_lg_boson()	21
5.11.1.3 subduce_lg_fermion()	22
5.11.1.4 subduce_oct()	22
5.12 SubdPol Namespace Reference	23
5.12.1 Function Documentation	23
5.12.1.1 Subduce_with_pol()	23
<b>6 Data Type Documentation</b>	<b>25</b>
6.1 hadron Struct Reference	25
6.1.1 Field Documentation	25
6.1.1.1 elab	25
6.1.1.2 ell	25
6.1.1.3 levels	26
6.1.1.4 max_mom	26
6.1.1.5 name	26
6.1.1.6 P	26
6.1.1.7 twoJ	26
6.2 irrep_label Struct Reference	26
6.2.1 Member Function Documentation	27
6.2.1.1 operator<()	27
6.2.2 Field Documentation	27
6.2.2.1 irrep	27
6.2.2.2 n	27
6.2.2.3 P	27
6.2.2.4 row	27
6.2.2.5 twoJ	27
6.3 KFacParams Class Reference	28
6.3.1 Constructor & Destructor Documentation	28
6.3.1.1 ~KFacParams()	28
6.3.1.2 KFacParams()	29
6.3.2 Member Function Documentation	29

6.3.2.1 subPhSum()	29
6.3.2.2 two_abs_lam()	30
6.3.3 Field Documentation	30
6.3.3.1 phase	30
6.3.3.2 qm	30
6.3.3.3 qp	30
6.3.3.4 Sub1	30
6.3.3.5 Sub3	30
6.3.3.6 SubCurr	31
6.4 KfacSSS Class Reference	31
6.4.1 Member Function Documentation	32
6.4.1.1 name()	32
6.4.1.2 operator()	32
6.5 KfacSSV Class Reference	32
6.5.1 Member Function Documentation	33
6.5.1.1 name()	33
6.5.1.2 operator()	33
6.6 KfacSVS Class Reference	34
6.6.1 Member Function Documentation	34
6.6.1.1 name()	35
6.6.1.2 operator()	35
6.7 KfacSVV Class Reference	35
6.7.1 Member Function Documentation	36
6.7.1.1 name()	36
6.7.1.2 operator()	36
6.8 KFactor Class Reference	37
6.8.1 Constructor & Destructor Documentation	37
6.8.1.1 ~KFactor()	37
6.8.2 Member Function Documentation	37
6.8.2.1 name()	37
6.8.2.2 operator()	38
6.9 KfacVSS Class Reference	38
6.9.1 Member Function Documentation	39
6.9.1.1 name()	39
6.9.1.2 operator()	39
6.10 KfacVVS Class Reference	40
6.10.1 Member Function Documentation	40
6.10.1.1 name()	40
6.10.1.2 operator()	41
6.11 Ph::phChars Struct Reference	41
6.11.1 Member Function Documentation	41
6.11.1.1 operator<()	42

6.11.2 Field Documentation	42
6.11.2.1 lam_phase	42
6.11.2.2 mom1	42
6.11.2.3 mom2	42
6.11.2.4 r	42
6.12 KfUt::ToArray Class Reference	42
6.12.1 Member Function Documentation	43
6.12.1.1 toArray() [1/2]	43
6.12.1.2 toArray() [2/2]	43
<b>7 File Documentation</b>	<b>45</b>
7.1 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_prefactor.cc File Reference	45
7.1.1 Function Documentation	45
7.1.1.1 main()	45
7.2 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.cc File Reference	46
7.2.1 Function Documentation	46
7.2.1.1 main()	46
7.3 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h File Reference	47
7.3.1 Function Documentation	47
7.3.1.1 write_ei()	48
7.4 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/xml_tools.cc File Reference	48
7.4.1 Function Documentation	48
7.4.1.1 write_ei()	48
7.5 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.cc File Reference	49
7.6 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h File Reference	49
7.7 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.cc File Reference	50
7.8 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h File Reference	51
7.9 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.cc File Reference	52
7.10 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.h File Reference	53
7.10.1 Typedef Documentation	54
7.10.1.1 TheKFactorFactory	54
7.11 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc File Reference	54
7.12 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h File Reference	54
7.13 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc File Reference	55
7.14 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h File Reference	56
7.15 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc File Reference	57
7.16 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h File Reference	57
7.17 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc File Reference	58
7.18 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h File Reference	58
7.19 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h File Reference	59
7.20 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.cc File Reference	60
7.21 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.h File Reference	61

7.22 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.cc File Reference . . . . .	62
7.23 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.h File Reference . . . . .	63
7.24 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc File Reference . . . . .	64
7.25 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h File Reference . . . . .	64
7.26 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.cc File Reference . . . . .	65
7.26.1 Function Documentation . . . . .	65
7.26.1.1 Round() . . . . .	65
7.27 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h File Reference . . . . .	65
7.27.1 Function Documentation . . . . .	66
7.27.1.1 Round() . . . . .	66
7.28 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.cc File Reference . . . . .	67
7.29 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.h File Reference . . . . .	67
7.30 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.cc File Reference . . . . .	68
7.31 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.h File Reference . . . . .	69
7.32 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.cc File Reference . . . . .	70
7.33 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.h File Reference . . . . .	70
7.34 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.cc File Reference . . . . .	71
7.34.1 Function Documentation . . . . .	71
7.34.1.1 linkageHack() . . . . .	71
7.35 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h File Reference . . . . .	72
<b>Index</b>	<b>75</b>





# Chapter 1

## Modules Index

### 1.1 Modules List

Here is a list of all modules with brief descriptions:

<a href="#">IrrepName</a>	9
<a href="#">iter</a>	10
<a href="#">KFactorEnv</a>	10
<a href="#">KfUt</a>	11
<a href="#">LevCiv</a>	13
<a href="#">LittleGrp</a>	13
<a href="#">naming</a>	14
<a href="#">Ph</a>	15
<a href="#">PolVec</a>	18
<a href="#">Rot</a>	20
<a href="#">Subd</a>	20
<a href="#">SubdPol</a>	23



## Chapter 2

# Data Type Index

### 2.1 Class Hierarchy

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

hadron . . . . .	25
irrep_label . . . . .	26
KFacParams . . . . .	28
KFactor . . . . .	37
KfacSSS . . . . .	31
KfacSSV . . . . .	32
KfacSVS . . . . .	34
KfacSVV . . . . .	35
KfacVSS . . . . .	38
KfacVVS . . . . .	40
Ph::phChars . . . . .	41
KfUt::ToArray . . . . .	42



## Chapter 3

# Data Type Index

### 3.1 Data Types List

Here are the data types with brief descriptions:

hadron	25
irrep_label	26
KFacParams	28
KfacSSS	31
KfacSSV	32
KfacSVS	34
KfacSVV	35
KFactor	37
KfacVSS	38
KfacVVS	40
Ph::phChars	41
KfUt::ToArray	42



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_prefactor.cc . . . . .	45
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.cc . . . . .	46
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h . . . . .	47
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/xml_tools.cc . . . . .	48
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.cc . . . . .	49
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h . . . . .	49
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.cc . . . . .	50
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h . . . . .	51
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.cc . . . . .	52
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.h . . . . .	53
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc . . . . .	54
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h . . . . .	54
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc . . . . .	55
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h . . . . .	56
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc . . . . .	57
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h . . . . .	57
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc . . . . .	58
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h . . . . .	58
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h . . . . .	59
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.cc . . . . .	60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.h . . . . .	61
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.cc . . . . .	62
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.h . . . . .	63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc . . . . .	64
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h . . . . .	64
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.cc . . . . .	65
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h . . . . .	65
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.cc . . . . .	67
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.h . . . . .	67
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.cc . . . . .	68
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.h . . . . .	69
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.cc . . . . .	70
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.h . . . . .	70
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.cc . . . . .	71
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h . . . . .	72





## Chapter 5

# Module Documentation

### 5.1 IrrepName Namespace Reference

#### Functions

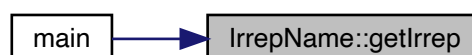
- `std::vector< std::string > getIrrep (int &twoJ, int &P, string &lg)`
- `int irrepRows (string &irrep)`

#### 5.1.1 Function Documentation

##### 5.1.1.1 `getIrrep()`

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

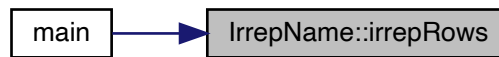
Here is the caller graph for this function:



### 5.1.1.2 irrepRows()

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

Here is the caller graph for this function:



## 5.2 iter Namespace Reference

### Functions

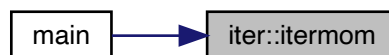
- `std::vector< Vector3d > itermom (double max_mom)`

### 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

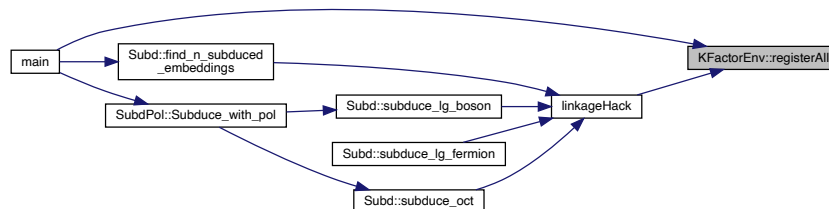
- `bool registerAll ()`

### 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

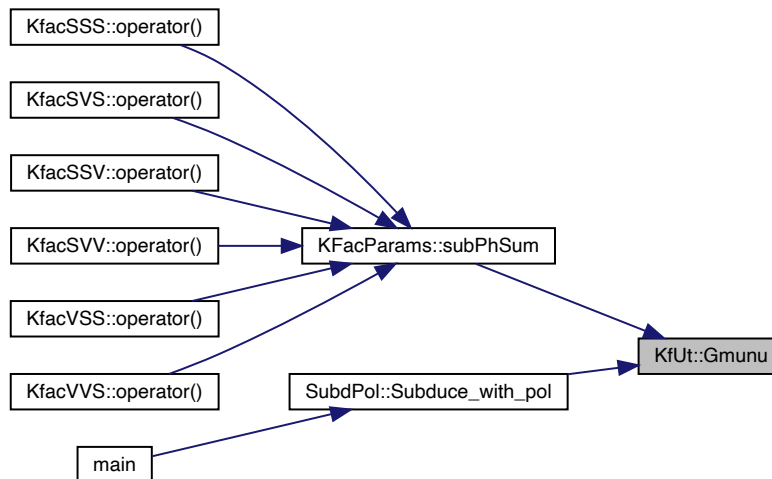
- double [truncate](#) (double num, int precision)
- `Eigen::MatrixXcd` [Gmunu](#) ()

#### 5.4.1 Function Documentation

#### 5.4.1.1 Gmunu()

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

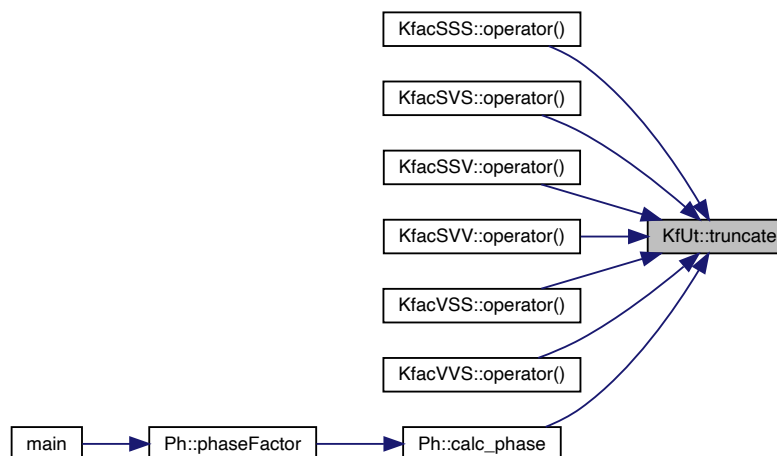
Here is the caller graph for this function:



#### 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

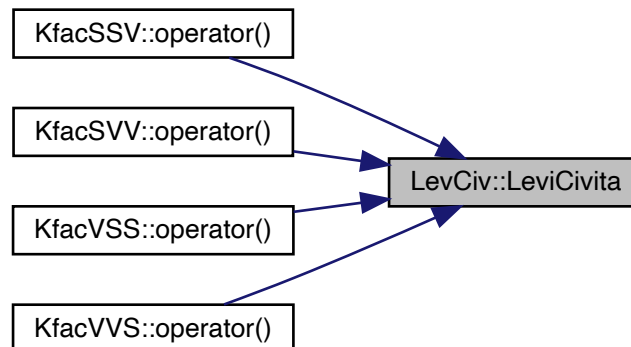
- double [LeviCivita](#) (int arr[], int n)

#### 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_ )
```

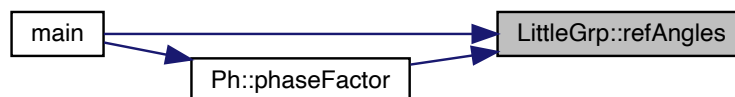
Here is the caller graph for this function:



### 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 LGL,
    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

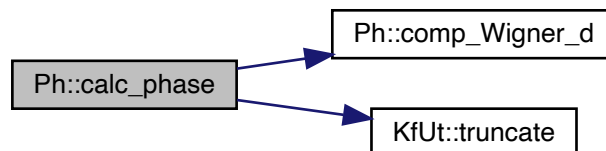
```
typedef std::tuple<int, int, int> Ph::tripKey
```

## 5.8.2 Function Documentation

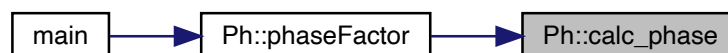
### 5.8.2.1 calc\_phase()

```
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:

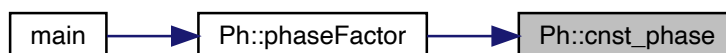




5.8.2.2 `cnst_phase()`

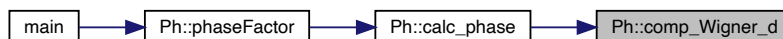
```
map< Ph::tripKey, complex< double > > Ph::cnst_phase (
    int twoJ1,
    int twoJ2,
    int twoJCurr )
```

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 )
```

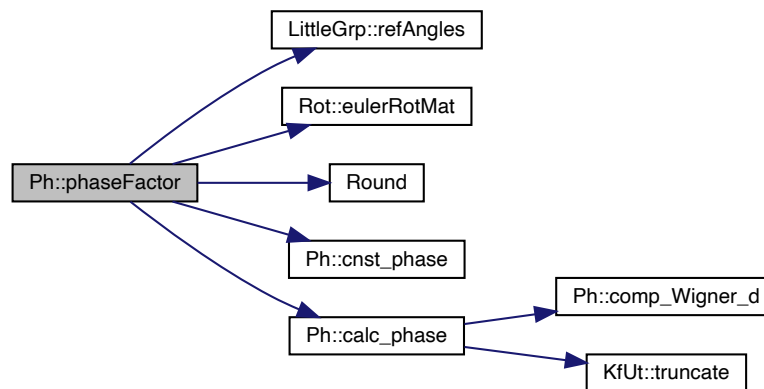
Here is the caller graph for this function:



#### 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

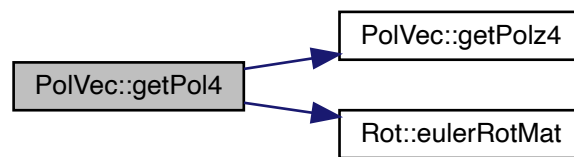
## 5.9.1.1 getPol4()

```

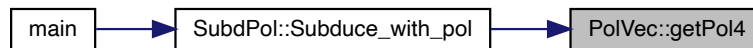
Eigen::MatrixXcd PolVec::getPol4 (
    double & mom_sq,
    const int & two_helicity,
    double & mass_sq,
    double & phi,
    double & theta,
    double & psi,
    bool curr )

```

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

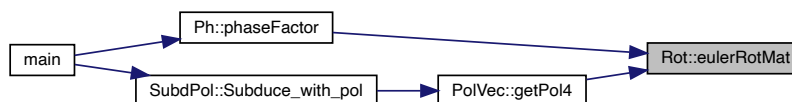
- Eigen::MatrixXd [eulerRotMat](#) (double alpha, double beta, double gamma)

### 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

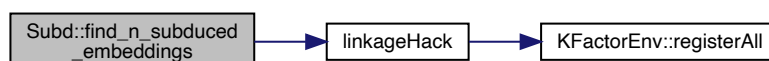
- map< int, complex< double > > [subduce\\_lg\\_boson](#) (const [irrep\\_label](#) &irrep, const string &little\_group)
- map< int, complex< double > > [subduce\\_lg\\_fermion](#) (const [irrep\\_label](#) &irrep, const string &little\_group)
- map< int, complex< double > > [subduce\\_oct](#) (const [irrep\\_label](#) &irrep)
- int [find\\_n\\_subduced\\_embeddings](#) (const string &group, const string &irrep, int twoJ, int eta\_tilde)

### 5.11.1 Function Documentation

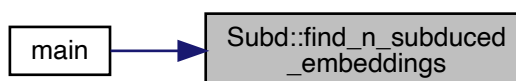
## 5.11.1.1 find\_n\_subduced\_embeddings()

```
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:



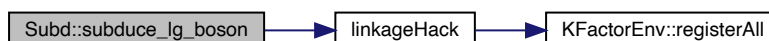
Here is the caller graph for this function:



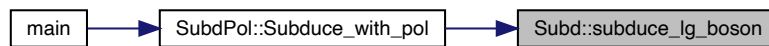
## 5.11.1.2 subduce\_lg\_boson()

```
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:

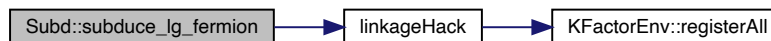


#### 5.11.1.3 subduce\_lg\_fermion()

```

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

Here is the call graph for this function:

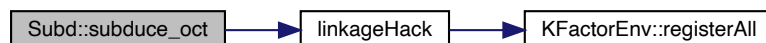


#### 5.11.1.4 subduce\_oct()

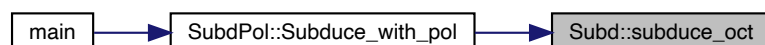
```

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

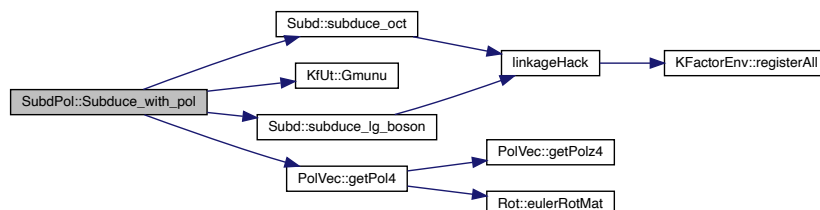
- `map< int, Eigen::MatrixXcd > Subduce_with_pol` (double &mom\_sq, double &mass\_sq, int &twoJ, const `irrep_label` &irrep, const string &little\_group, double R1\_phi, double R1\_theta, double R1\_psi, bool curr)

### 5.12.1 Function Documentation

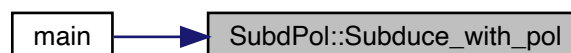
#### 5.12.1.1 Subduce\_with\_pol()

```
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
```

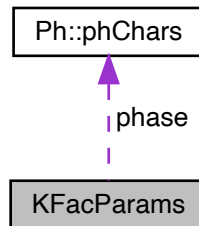
The documentation for this struct was generated from the following file:

- /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_ )

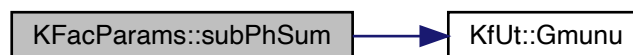
```

## 6.3.2 Member Function Documentation

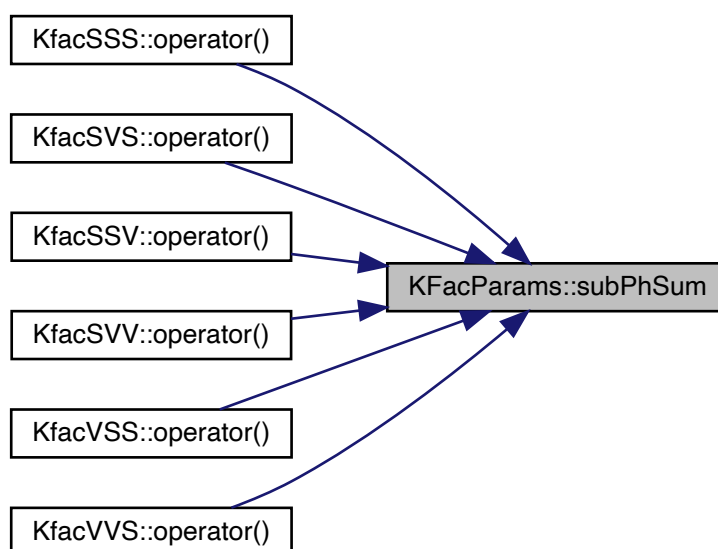
## 6.3.2.1 subPhSum()

```
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
```

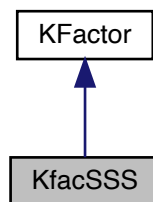
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/[kfac\\_params.h](#)
- /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/[kfac\\_params.cc](#)

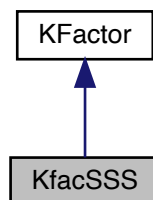
## 6.4 KfacSSS Class Reference

```
#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]
```

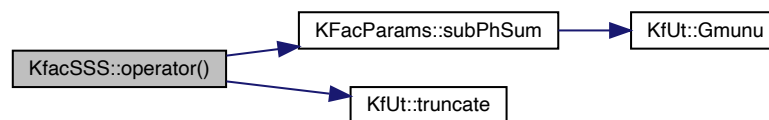
Implements [KFactor](#).

### 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:



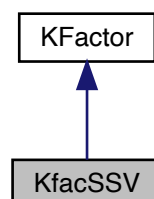
The documentation for this class was generated from the following files:

- `/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

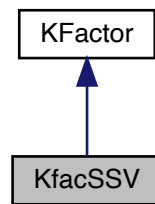
```
#include <kfactor_pigammarho.h>
```

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]
```

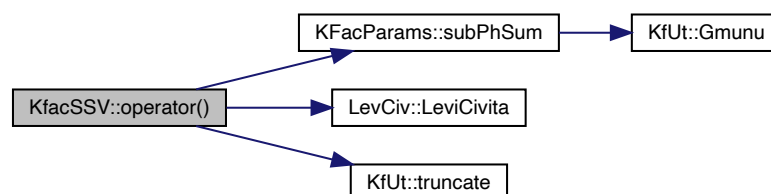
Implements [KFactor](#).

#### 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:



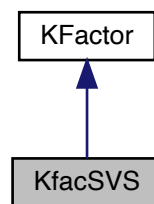
The documentation for this class was generated from the following files:

- [/Users/archanar/git/Kinematic\\_factors/compute\\_Kfactors/src/lib/kfactor\\_pigammarho.h](/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h)
- [/Users/archanar/git/Kinematic\\_factors/compute\\_Kfactors/src/lib/kfactor\\_pigammarho.cc](/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc)

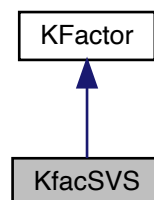
## 6.6 KfacSVS Class Reference

```
#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]
```

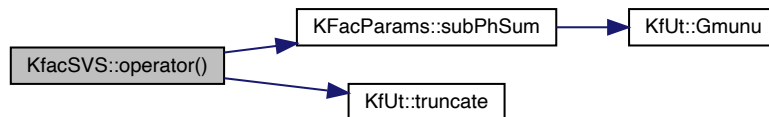
Implements [KFactor](#).

## 6.6.1.2 operator&gt;()

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

Implements [KFactor](#).

Here is the call graph for this function:



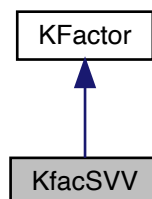
The documentation for this class was generated from the following files:

- [/Users/archanar/git/Kinematic\\_factors/compute\\_Kfactors/src/lib/kfactor\\_pigammapi.h](#)
- [/Users/archanar/git/Kinematic\\_factors/compute\\_Kfactors/src/lib/kfactor\\_pigammapi.cc](#)

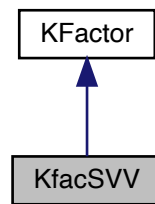
## 6.7 KfacSVV Class Reference

```
#include <kfactor_pigammapi.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]
```

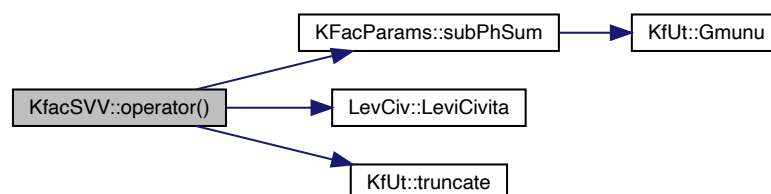
Implements [KFactor](#).

#### 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:



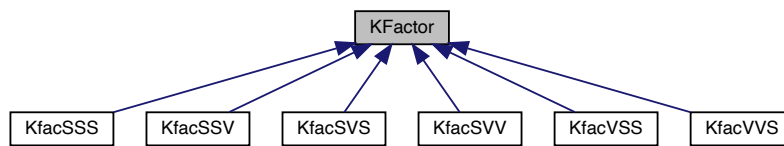
The documentation for this class was generated from the following files:

- [/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](#).

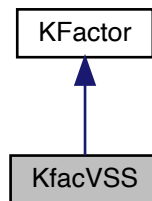
The documentation for this class was generated from the following file:

- [/Users/archanar/git/Kinematic\\_factors/compute\\_Kfactors/src/lib/kfactor\\_pigammarho.h](#)

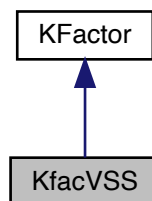
## 6.9 KfacVSS Class Reference

```
#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]
```

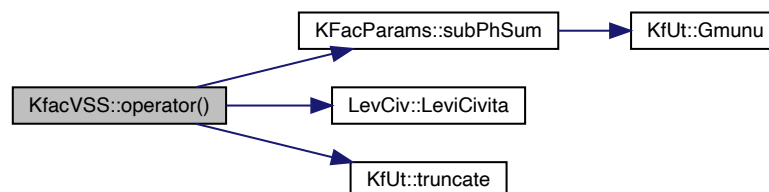
Implements [KFactor](#).

### 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:



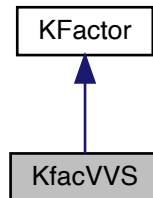
The documentation for this class was generated from the following files:

- `/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor\_pigammarho.h`
- `/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor\_pigammarho.cc`

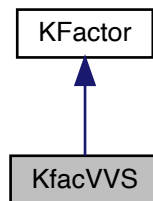
## 6.10 KfacVVS Class Reference

```
#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](#).

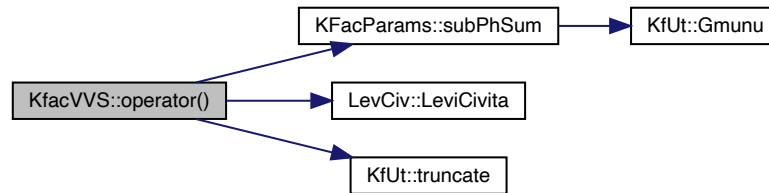


## 6.10.1.2 operator()

```
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](#)



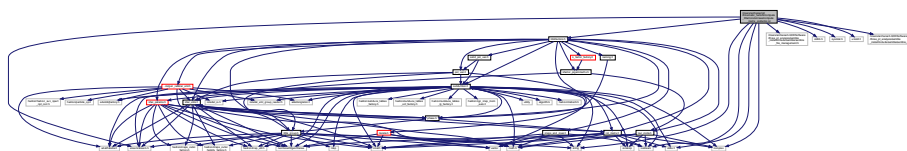
## Chapter 7

# 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

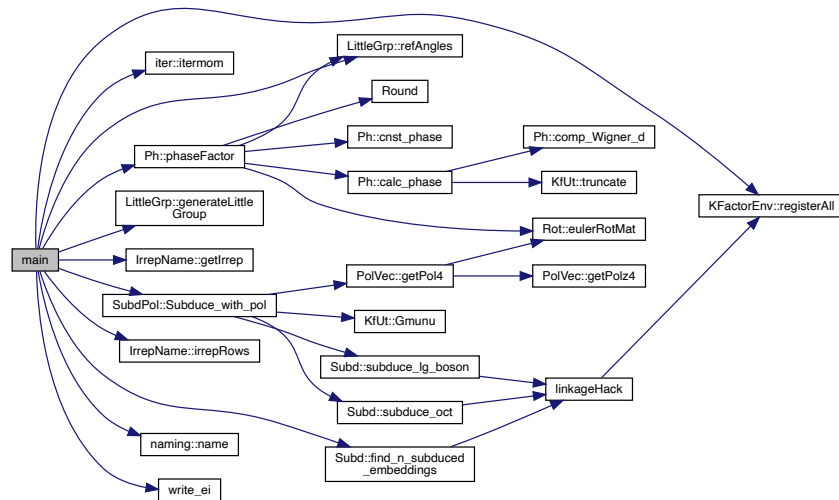
- int [main](#) (int argc, char \*\*argv)

#### 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

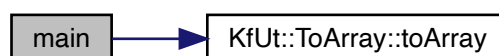
- int [main](#) (int argc, char \*\*argv)

#### 7.2.1 Function Documentation

##### 7.2.1.1 main()

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

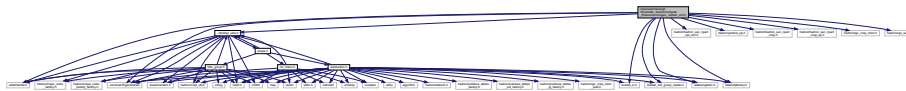
Here is the call graph for this function:



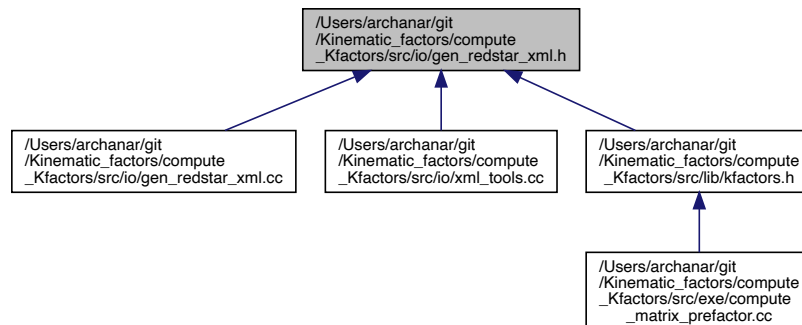
## 7.3 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/io/gen\_redstar\_xml.h File Reference

```
#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

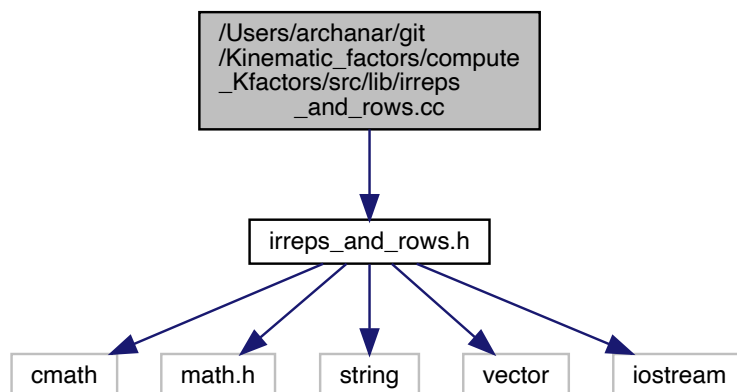




## 7.5 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/irreps\_and\_rows.cc File Reference

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

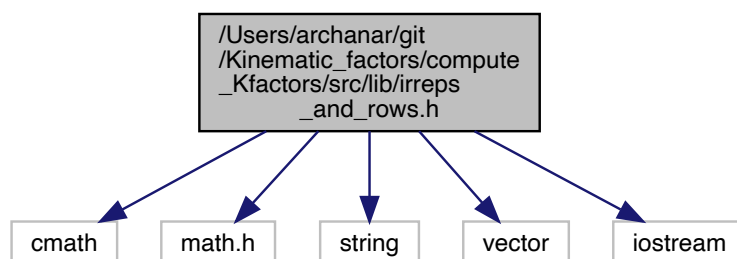
Include dependency graph for irreps\_and\_rows.cc:



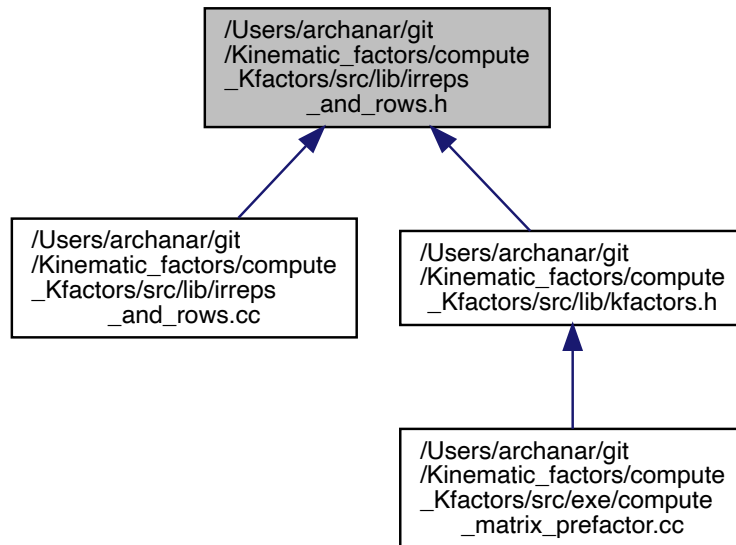
## 7.6 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/irreps\_and\_rows.h File Reference

```
#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](#)

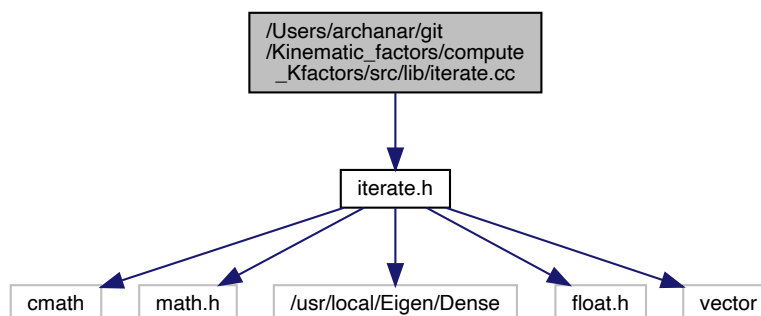
## Functions

- `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"
```

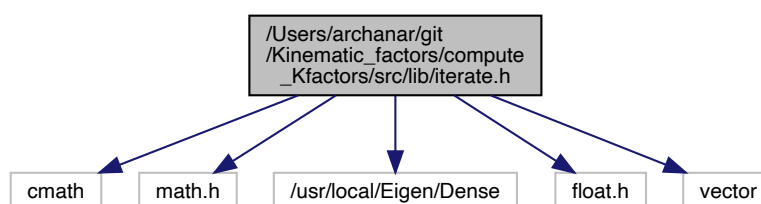
Include dependency graph for iterate.cc:



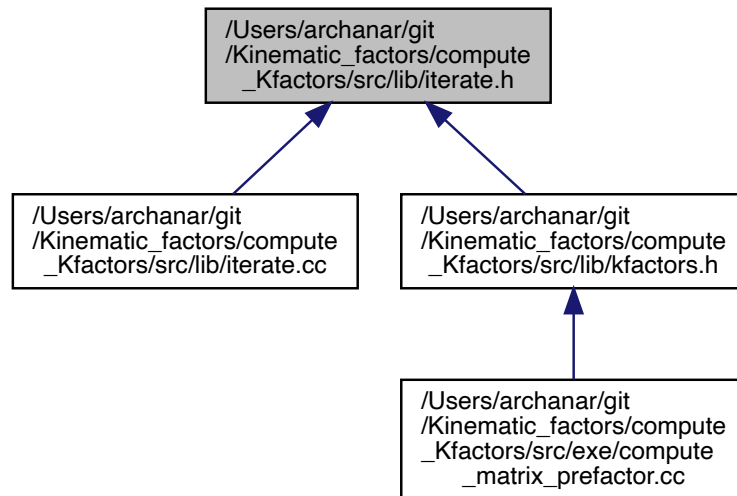
## 7.8 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/iterate.h File Reference

```
#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

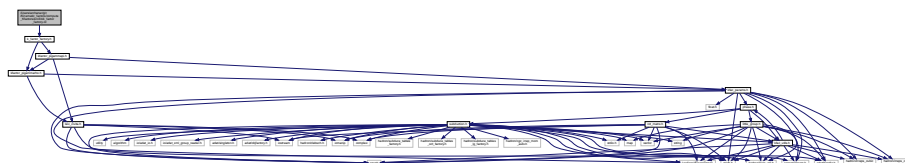
- `std::vector< Vector3d > iter::itermom (double max_mom)`

## 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

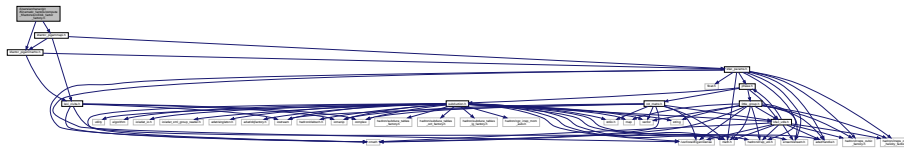
- [KFactorEnv](#)

## Functions

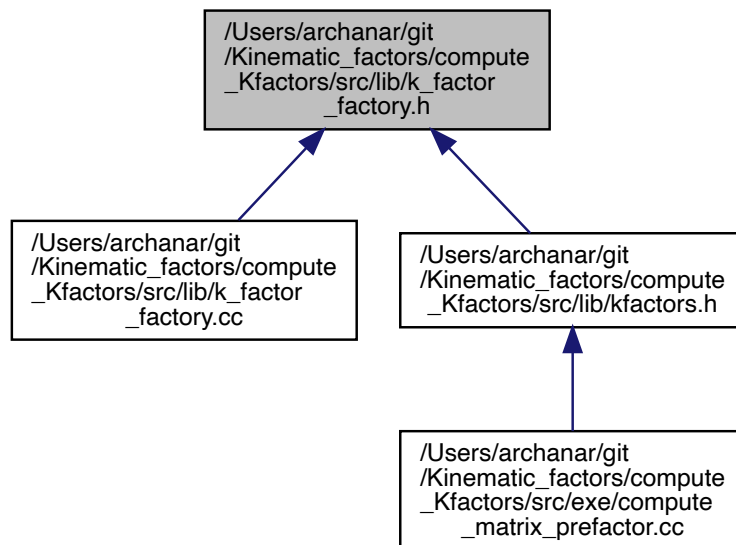
- bool [KFactorEnv::registerAll](#) ()

## 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

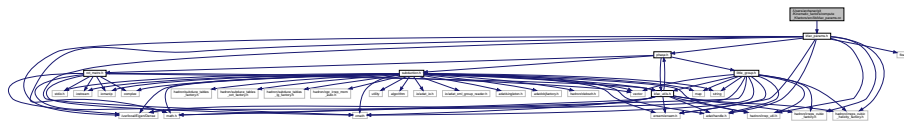
#### 7.10.1.1 TheKFactorFactory

```
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"
```

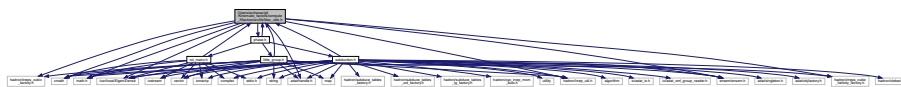
Include dependency graph for kfac\_params.h:



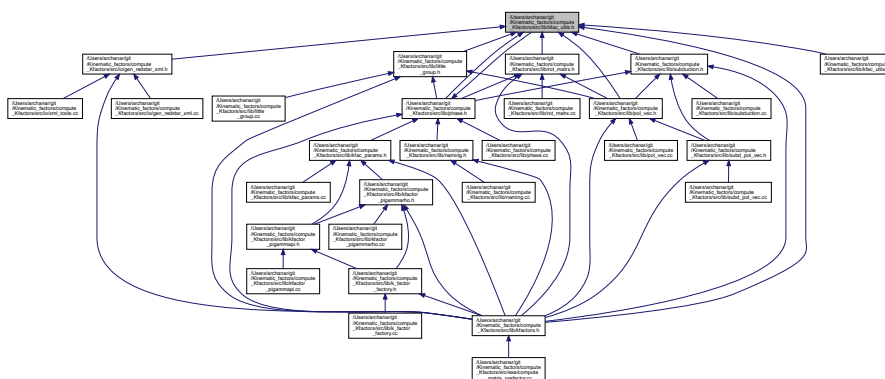


7.14 `/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h` [File](#)  
[Reference](#)

```
#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 kfacs_utils.h:
```



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



## Data Structures

- class `KfUt::ToArray`

## Namespaces

- KfUt

## Functions

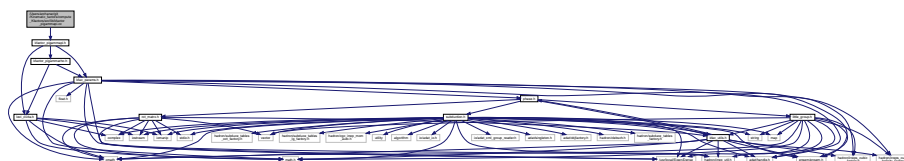
- double `KfUt::truncate` (double num, int precision)
- `Eigen::MatrixXcd KfUt::Gmunu` ()



## 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:



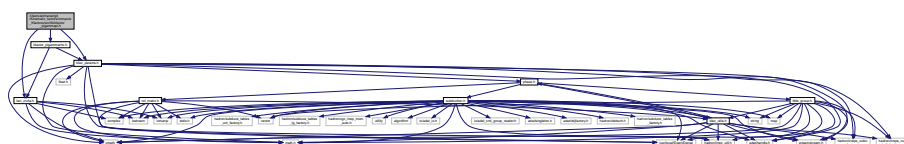
## 7.16 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h File Reference

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

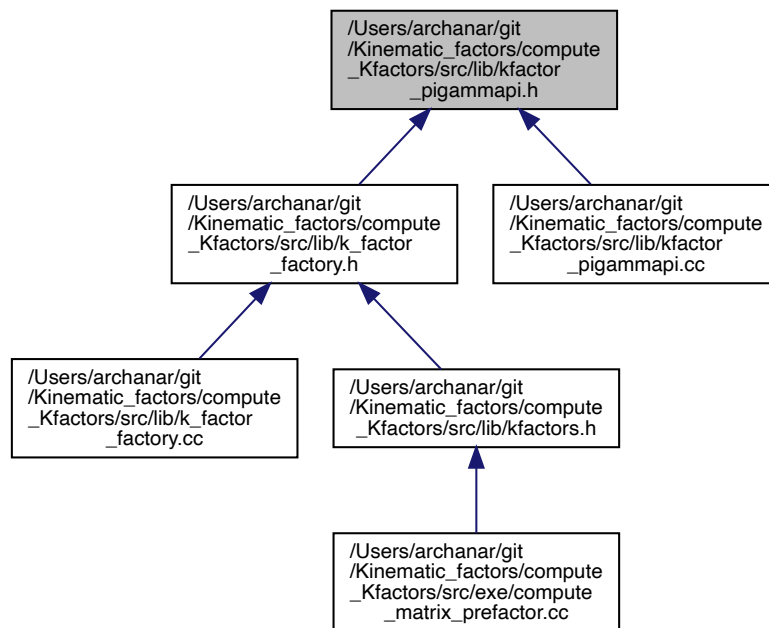
```
#include "levi_civita.h"
```

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

Include dependency graph for kfactor\_pigammapi.h:



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



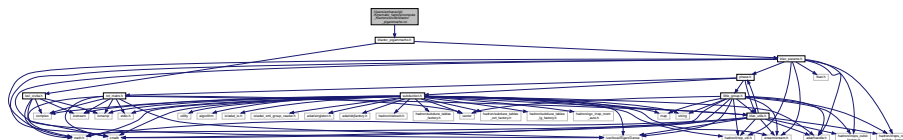
## Data Structures

- class [KfacSVS](#)
- class [KfacSSS](#)

## 7.17 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h File Reference

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

Include dependency graph for kfactor\_pigammapi.cc:

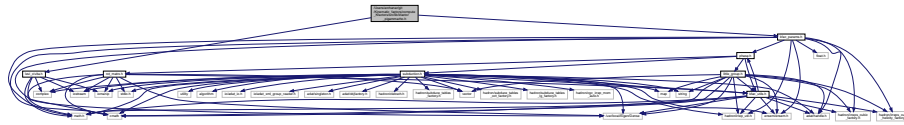


## 7.18 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h File Reference

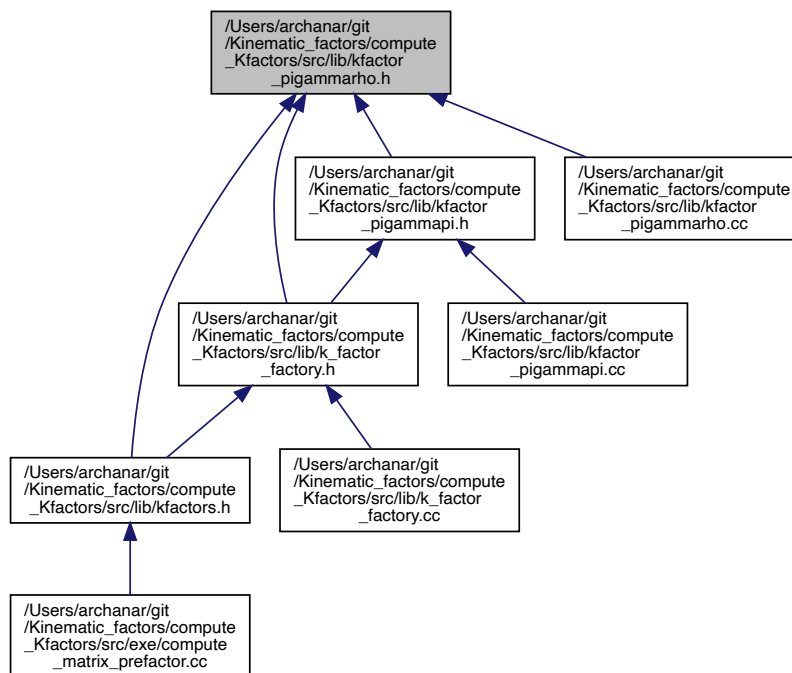
```
#include "kfac_params.h"
```

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

Include dependency graph for kfactor\_pigammarho.h:



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



## Data Structures

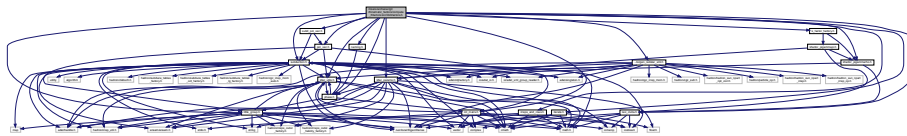
- class [KFactor](#)
- class [KfacSVV](#)
- class [KfacSSV](#)
- class [KfacVVS](#)
- class [KfacVSS](#)

## 7.19 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactors.h File Reference

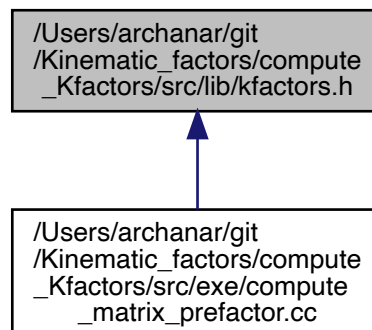
```
#include "subduction.h"
#include "subd_pol_vec.h"
#include "pol_vec.h"
```

```
#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:



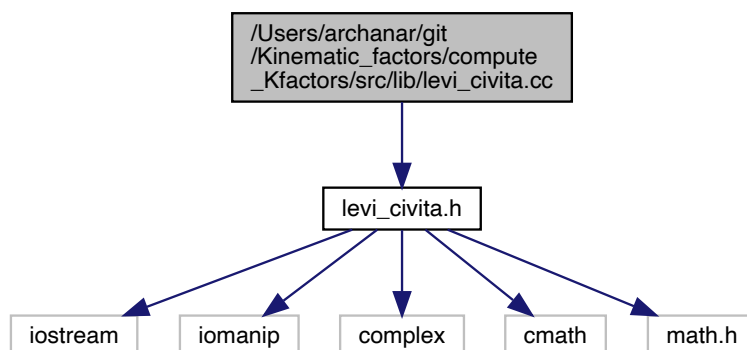
This graph shows which files directly or indirectly include this file:



## 7.20 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.cc File Reference

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

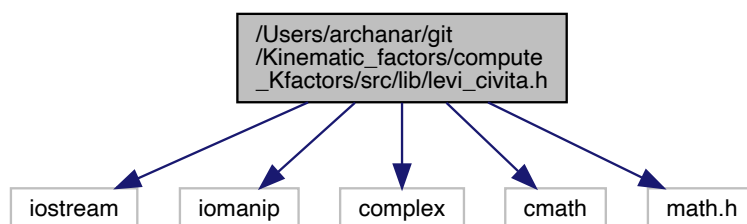
Include dependency graph for levi\_civita.cc:



## 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:





## Functions

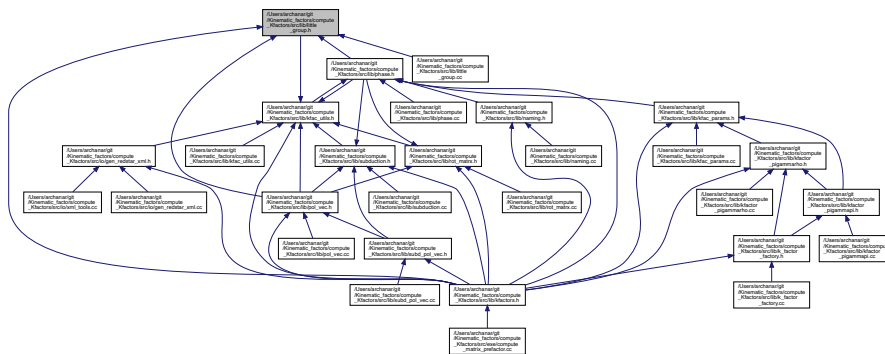
- string [LittleGrp::generateLittleGroup](#) (Eigen::Vector3d &mom\_)
- std::vector< double > [LittleGrp::refAngles](#) (Eigen::Vector3d mom1)

## 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

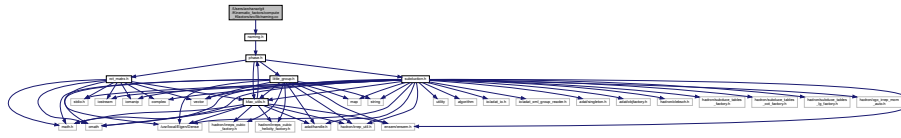
- [LittleGrp](#)

## Functions

- string [LittleGrp::generateLittleGroup](#) (Eigen::Vector3d &mom\_)
- std::vector< double > [LittleGrp::refAngles](#) (Eigen::Vector3d mom1)

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

Include dependency graph for naming.cc:



```
#include "phase.h"
Include dependency graph for naming.h:
```



```
graph BT; A["/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h"] --> B["/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h"]; A --> C["/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc"]; B --> D["/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_prefactor.cc"];
```

The diagram illustrates the following dependencies:

- `naming.h` depends on `kfactors.h` and `naming.cc`.
- `compute_matrix_prefactor.cc` depends on `kfactors.h`.

- naming

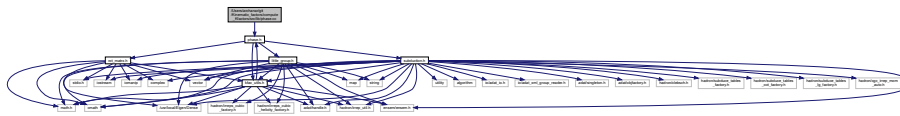


## Functions

- 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()

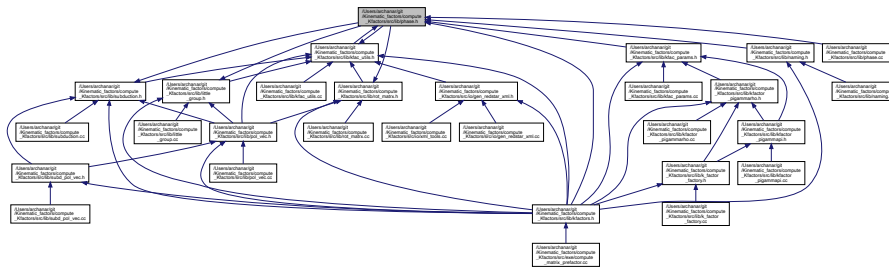
```
double Round (
    double x )
```

## 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<sub>←</sub>\_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<sub>←</sub>\_mom\_curr)
- map< [Ph::tripKey](#), complex< double > > [Ph::cnst\\_phase](#) (int twoJ1, int twoJ2, int twoJCurr)

### 7.27.1 Function Documentation

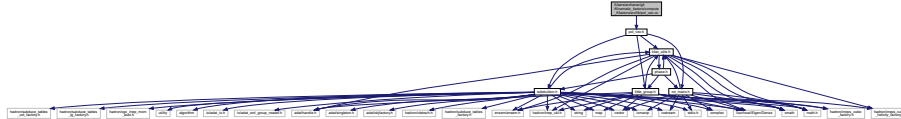
#### 7.27.1.1 Round()

```
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](#)

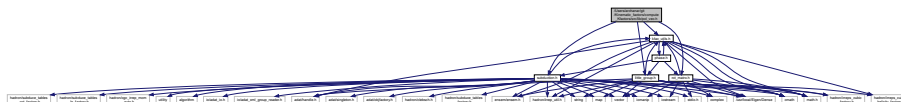
### Functions

- 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)

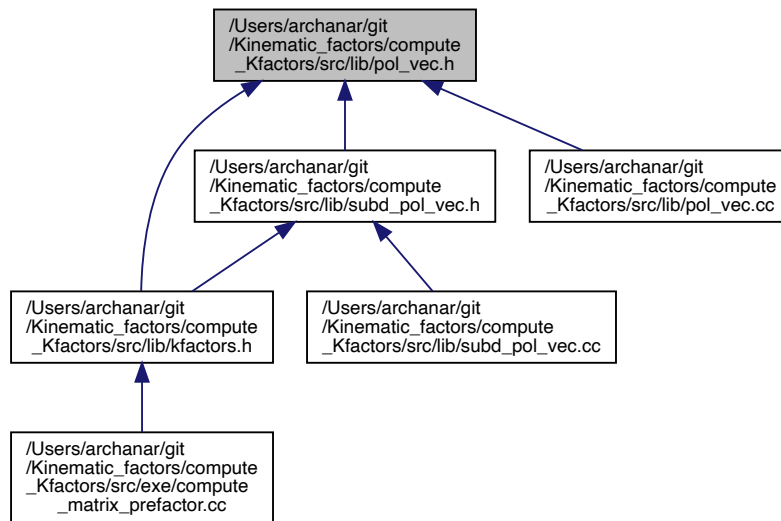
## 7.29 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/pol\_vec.h File Reference

```
#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)

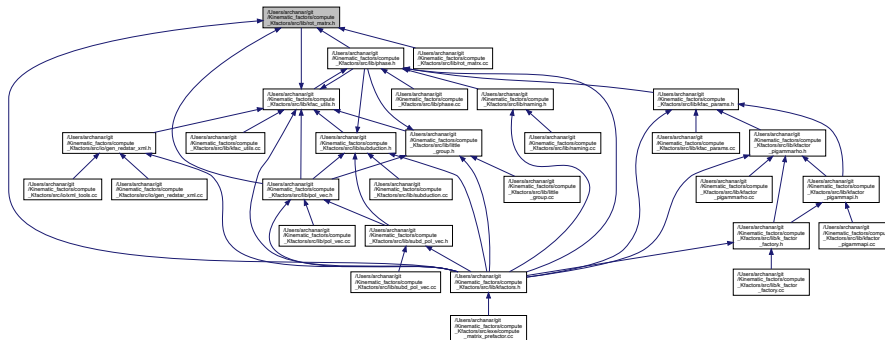
## 7.31 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/rot\_mtx.h File Reference

```
#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\_mtx.h:



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



## Namespaces

- [Rot](#)

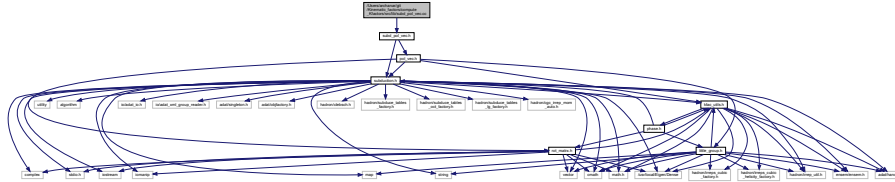
## Functions

- 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:

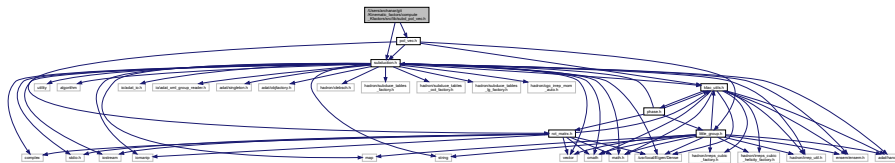


### 7.33 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/subd\_pol\_vec.h File Reference

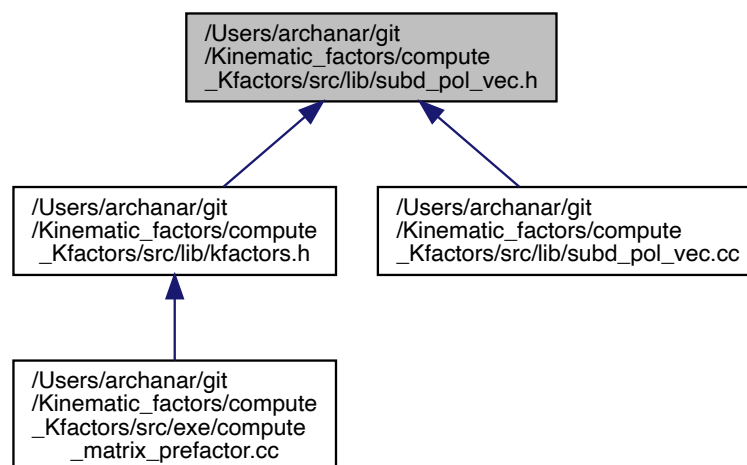
```
#include "subduction.h"
```

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

Include dependency graph for subd\_pol\_vec.h:



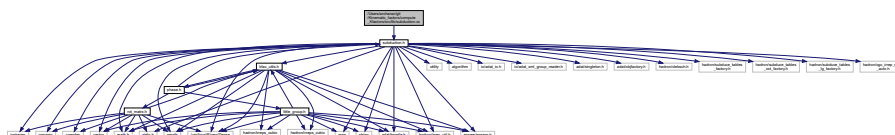
This graph shows which files directly or indirectly include this file:



- SubdPol

- `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)`

Include dependency graph for subduction.cc:



- bool `linkageHack` (void)

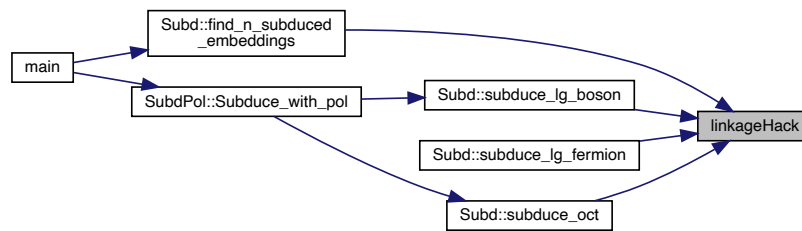
### 7.34.1.1 linkageHack()

```
bool linkageHack (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



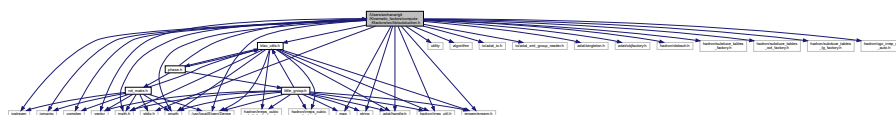
### 7.35 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/subduction.h File Reference

```

#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:





- struct `irrep_label`

- Subd

- `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

/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/exe/compute\_matrix\_prefactor.cc,  
45 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/phase.cc,  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/io/gen\_redstar\_xml.cc,  
46 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/phase.h,  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/io/gen\_redstar\_xml.h,  
47 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/pol\_vec.cc,  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/io/xml\_tools.cc,  
48 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/pol\_vec.h,  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/irreps\_and\_rows.cc,  
49 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/rot\_matrix.  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/irreps\_and\_rows.h,  
49 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/rot\_matrix.  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/iterate.cc,  
50 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/subd\_pol.  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/iterate.h,  
51 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/subd\_pol.  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/k\_factor\_factory.cc,  
52 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/subduction.  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/k\_factor\_factory.h,  
53 /Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/subduction.  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfac\_params.cc,  
54 ~KFacParams  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfac\_params.h,  
54 ~KFactor  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_37,  
55 KFactor  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfac\_utils.cc,  
56 calc\_phase  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfac\_utils.h,  
56 Ph, 16  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.cc,  
57 cnst\_phase  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h,  
57 Ph, 17  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h,  
57 comp\_Wigner\_d  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h,  
57 Ph, 17  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h,  
57 compute\_matrix\_prefactor.cc  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h,  
58 main, 45  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactor\_pigammapi.h,  
58 erab  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/kfactors.h,  
59 hadron, 25  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.cc,  
60 euler\_RotMat  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
61 Rot, 20  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
61 find\_n\_subduced\_embeddings  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
62 title\_group.cc,  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
63 title\_group.cc  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
63 main, 46  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
64 write\_ei, 47  
/Users/archanar/git/Kinematic\_factors/compute\_Kfactors/src/lib/levi\_civita.h,  
64 title\_group

- LittleGrp, 13
- getIrrep
  - IrrepName, 9
- getPol4
  - PolVec, 18
- getPolz4
  - PolVec, 19
- Gmunu
  - KfUt, 11
- hadron, 25
  - elab, 25
  - ell, 25
  - levels, 25
  - max\_mom, 26
  - name, 26
  - P, 26
  - twoJ, 26
- irrep
  - irrep\_label, 27
- irrep\_label, 26
  - irrep, 27
  - n, 27
  - operator<, 27
  - P, 27
  - row, 27
  - twoJ, 27
- IrrepName, 9
  - getIrrep, 9
  - irrepRows, 9
- irrepRows
  - IrrepName, 9
- iter, 10
  - itermom, 10
- itermom
  - iter, 10
- k\_factor\_factory.h
  - TheKFactorFactory, 54
- KFacParams, 28
  - ~KFacParams, 28
  - KFacParams, 28
  - phase, 30
  - qm, 30
  - qp, 30
  - Sub1, 30
  - Sub3, 30
  - SubCurr, 30
  - subPhSum, 29
  - two\_abs\_lam, 30
- KfacSSS, 31
  - name, 32
  - operator(), 32
- KfacSSV, 32
  - name, 33
  - operator(), 33
- KfacSVS, 34
  - name, 34
- operator(), 35
- KfacSVV, 35
  - name, 36
  - operator(), 36
- KFactor, 37
  - ~KFactor, 37
  - name, 37
  - operator(), 37
- KFactorEnv, 10
  - registerAll, 11
- KfacVSS, 38
  - name, 39
  - operator(), 39
- KfacVVS, 40
  - name, 40
  - operator(), 40
- KfUt, 11
  - Gmunu, 11
  - truncate, 12
- KfUt::ToArray, 42
  - toArray, 43
- lam\_phase
  - Ph::phChars, 42
- LevCiv, 13
  - LeviCivita, 13
- levels
  - hadron, 25
- LeviCivita
  - LevCiv, 13
- linkageHack
  - subduction.cc, 71
- LittleGrp, 13
  - generateLittleGroup, 13
  - refAngles, 14
- main
  - compute\_matrix\_prefactor.cc, 45
  - gen\_redstar\_xml.cc, 46
- max\_mom
  - hadron, 26
- mom1
  - Ph::phChars, 42
- mom2
  - Ph::phChars, 42
- n
  - irrep\_label, 27
- name
  - hadron, 26
  - KfacSSS, 32
  - KfacSSV, 33
  - KfacSVS, 34
  - KfacSVV, 36
  - KFactor, 37
  - KfacVSS, 39
  - KfacVVS, 40
  - naming, 14
- naming, 14

- name, [14](#)
- operator<
  - irrep\_label, [27](#)
  - Ph::phChars, [41](#)
- operator()
  - KfacSSS, [32](#)
  - KfacSSV, [33](#)
  - KfacSVS, [35](#)
  - KfacSVV, [36](#)
  - KFactor, [37](#)
  - KfacVSS, [39](#)
  - KfacVVS, [40](#)
- P
  - hadron, [26](#)
  - irrep\_label, [27](#)
- Ph, [15](#)
  - calc\_phase, [16](#)
  - cnst\_phase, [16](#)
  - comp\_Wigner\_d, [17](#)
  - phaseFactor, [17](#)
  - tripKey, [16](#)
- Ph::phChars, [41](#)
  - lam\_phase, [42](#)
  - mom1, [42](#)
  - mom2, [42](#)
  - operator<, [41](#)
  - r, [42](#)
- phase
  - KFacParams, [30](#)
- phase.cc
  - Round, [65](#)
- phase.h
  - Round, [66](#)
- phaseFactor
  - Ph, [17](#)
- PolVec, [18](#)
  - getPol4, [18](#)
  - getPolz4, [19](#)
- qm
  - KFacParams, [30](#)
- qp
  - KFacParams, [30](#)
- r
  - Ph::phChars, [42](#)
- refAngles
  - LittleGrp, [14](#)
- registerAll
  - KFactorEnv, [11](#)
- Rot, [20](#)
  - eulerRotMat, [20](#)
- Round
  - phase.cc, [65](#)
  - phase.h, [66](#)
- row
  - irrep\_label, [27](#)
- Sub1
  - KFacParams, [30](#)
- Sub3
  - KFacParams, [30](#)
- SubCurr
  - KFacParams, [30](#)
- Subd, [20](#)
  - find\_n\_subduced\_embeddings, [20](#)
  - subduce\_lg\_boson, [21](#)
  - subduce\_lg\_fermion, [22](#)
  - subduce\_oct, [22](#)
- SubdPol, [23](#)
  - Subduce\_with\_pol, [23](#)
- subduce\_lg\_boson
  - Subd, [21](#)
- subduce\_lg\_fermion
  - Subd, [22](#)
- subduce\_oct
  - Subd, [22](#)
- Subduce\_with\_pol
  - SubdPol, [23](#)
- subduction.cc
  - linkageHack, [71](#)
- subPhSum
  - KFacParams, [29](#)
- TheKFactorFactory
  - k\_factor\_factory.h, [54](#)
- toArray
  - KfUt::ToArray, [43](#)
- tripKey
  - Ph, [16](#)
- truncate
  - KfUt, [12](#)
- two\_abs\_lam
  - KFacParams, [30](#)
- twoJ
  - hadron, [26](#)
  - irrep\_label, [27](#)
- write\_ei
  - gen\_redstar\_xml.h, [47](#)
  - xml\_tools.cc, [48](#)
- xml\_tools.cc
  - write\_ei, [48](#)