

Kinematic Factor

1.0

Generated by Doxygen 1.8.15

1 Modules Index	1
1.1 Modules List	1
2 Data Type Index	3
2.1 Class Hierarchy	3
3 Data Type Index	5
3.1 Data Types List	5
4 File Index	7
4.1 File List	7
5 Module Documentation	9
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	11
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	19
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
6 Data Type Documentation	25
6.1 flavour Struct Reference	25
6.1.1 Field Documentation	25
6.1.1.1 threeY	25
6.1.1.2 twol	25
6.1.1.3 twolz	25
6.2 irrep_label Struct Reference	26
6.2.1 Member Function Documentation	26
6.2.1.1 operator<()	26
6.2.2 Field Documentation	26
6.2.2.1 irrep	26
6.2.2.2 n	26
6.2.2.3 P	27
6.2.2.4 row	27
6.2.2.5 twoJ	27
6.3 IrrepLam_t Struct Reference	27
6.3.1 Field Documentation	27
6.3.1.1 irrep	27
6.3.1.2 lev	28
6.3.1.3 mom	28
6.3.1.4 mom_sq	28
6.3.1.5 row	28
6.3.1.6 two_lam	28
6.4 KFacParams Class Reference	28

6.4.1 Constructor & Destructor Documentation	29
6.4.1.1 ~KFacParams()	29
6.4.1.2 KFacParams()	29
6.4.2 Member Function Documentation	29
6.4.2.1 subPhSum()	30
6.4.2.2 two_abs_lam()	30
6.4.3 Field Documentation	30
6.4.3.1 phase	31
6.4.3.2 qm	31
6.4.3.3 qp	31
6.4.3.4 Sub1	31
6.4.3.5 Sub3	31
6.4.3.6 SubCurr	31
6.5 KfacSSS Class Reference	32
6.5.1 Member Function Documentation	32
6.5.1.1 name()	32
6.5.1.2 operator()()	33
6.6 KfacSSV Class Reference	33
6.6.1 Member Function Documentation	34
6.6.1.1 name()	34
6.6.1.2 operator()()	34
6.7 KfacSVS Class Reference	35
6.7.1 Member Function Documentation	35
6.7.1.1 name()	36
6.7.1.2 operator()()	36
6.8 KfacSVV Class Reference	36
6.8.1 Member Function Documentation	37
6.8.1.1 name()	37
6.8.1.2 operator()()	37
6.9 KFactor Class Reference	38
6.9.1 Constructor & Destructor Documentation	38
6.9.1.1 ~KFactor()	38
6.9.2 Member Function Documentation	38
6.9.2.1 name()	38
6.9.2.2 operator()()	39
6.10 KfacVSS Class Reference	39
6.10.1 Member Function Documentation	40
6.10.1.1 name()	40
6.10.1.2 operator()()	40
6.11 KfacVVS Class Reference	41
6.11.1 Member Function Documentation	41
6.11.1.1 name()	41

6.11.1.2 operator()	42
6.12 NPtCorr_t Struct Reference	42
6.12.1 Field Documentation	43
6.12.1.1 canonical	43
6.12.1.2 creation_op	43
6.12.1.3 elab	43
6.12.1.4 ell	43
6.12.1.5 flavor	44
6.12.1.6 levels	44
6.12.1.7 max_mom	44
6.12.1.8 min_mom	44
6.12.1.9 name	44
6.12.1.10 omit_mom	44
6.12.1.11 P	44
6.12.1.12 projected	44
6.12.1.13 smearedP	45
6.12.1.14 t_slice	45
6.12.1.15 twoJ	45
6.13 NPtLrepLam_t Struct Reference	45
6.13.1 Field Documentation	45
6.13.1.1 kfac	45
6.13.1.2 Npt	45
6.14 Ph::phChars Struct Reference	46
6.14.1 Member Function Documentation	46
6.14.1.1 operator<()	46
6.14.2 Field Documentation	46
6.14.2.1 lam_phase	46
6.14.2.2 mom1	46
6.14.2.3 mom2	46
6.14.2.4 r	47
6.15 KfUt::ToArray Class Reference	47
6.15.1 Member Function Documentation	47
6.15.1.1 toArray() [1/2]	47
6.15.1.2 toArray() [2/2]	47
7 File Documentation	49
7.1 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_prefactor.cc File Reference	49
7.1.1 Function Documentation	49
7.1.1.1 main()	49
7.2 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.cc File Reference	50
7.2.1 Function Documentation	50

7.2.1.1	read_xml_ini()	51
7.2.1.2	write_ei()	51
7.2.1.3	write_irrep()	51
7.2.1.4	write_xml_out()	52
7.3	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.h File Reference	52
7.3.1	Function Documentation	53
7.3.1.1	read_xml_ini()	54
7.3.1.2	write_ei()	54
7.3.1.3	write_irrep()	54
7.3.1.4	write_xml_out()	55
7.4	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.cc File Reference	55
7.4.1	Function Documentation	55
7.4.1.1	gen_redstar_xml()	56
7.5	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h File Reference	56
7.5.1	Function Documentation	57
7.5.1.1	gen_redstar_xml()	57
7.6	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.cc File Reference	57
7.6.1	Function Documentation	57
7.6.1.1	compute_three_point_prefactor()	58
7.7	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.h File Reference	58
7.7.1	Function Documentation	59
7.7.1.1	compute_three_point_prefactor()	59
7.8	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.cc File Reference	60
7.9	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h File Reference	60
7.10	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.cc File Reference	61
7.11	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h File Reference	62
7.12	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.cc File Reference	63
7.13	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.h File Reference	63
7.13.1	Typedef Documentation	64
7.13.1.1	TheKFactorFactory	64
7.14	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc File Reference	65
7.15	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h File Reference	65
7.16	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc File Reference	66
7.17	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h File Reference	67
7.18	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc File Reference	68
7.19	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h File Reference	68
7.20	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc File Reference	69
7.21	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h File Reference	70
7.22	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h File Reference	71
7.23	/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.cc File Reference	72

7.24 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.h File Reference	72
7.25 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.cc File Reference	73
7.26 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.h File Reference	74
7.27 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc File Reference	75
7.28 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h File Reference	75
7.29 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.cc File Reference	76
7.29.1 Function Documentation	77
7.29.1.1 Round()	77
7.30 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h File Reference	77
7.30.1 Function Documentation	78
7.30.1.1 Round()	78
7.31 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.cc File Reference	78
7.32 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.h File Reference	79
7.33 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.cc File Reference	80
7.34 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.h File Reference	80
7.35 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.cc File Reference	81
7.36 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.h File Reference	81
7.37 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.cc File Reference	82
7.37.1 Function Documentation	83
7.37.1.1 linkageHack()	83
7.38 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h File Reference	83
Index	85

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

IrrepName	9
iter	10
KFactorEnv	11
KfUt	11
LevCiv	13
LittleGrp	13
naming	14
Ph	15
PolVec	18
Rot	20
Subd	20
SubdPol	23

Chapter 2

Data Type Index

2.1 Class Hierarchy

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

flavour	25
irrep_label	26
IrrepLam_t	27
KFacParams	28
KFactor	38
KfacSSS	32
KfacSSV	33
KfacSVS	35
KfacSVV	36
KfacVSS	39
KfacVVS	41
NPtCorr_t	42
NPtIrrepLam_t	45
Ph::phChars	46
KfUt::ToArray	47

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

flavour	25
irrep_label	26
IrrepLam_t	27
KFacParams	28
KfacSSS	32
KfacSSV	33
KfacSVS	35
KfacSVV	36
KFactor	38
KfacVSS	39
KfacVVS	41
NPtCorr_t	42
NPtIrrepLam_t	45
Ph::phChars	46
KfUt::ToArray	47

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	49
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.cc . . .	50
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.h . . .	52
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.cc	55
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h	56
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.cc . . .	57
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.h . . .	58
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.cc	60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h	60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.cc	61
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h	62
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.cc	63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.h	63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc	65
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h	65
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc	66
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h	67
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc	68
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h	68
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc	69
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h	70
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h	71
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.cc	72
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/levi_civita.h	72
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.cc	73
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.h	74
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc	75
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h	75
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.cc	76
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h	77
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.cc	78
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.h	79
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.cc	80
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrx.h	80
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.cc	81
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.h	81
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.cc	82
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h	83

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

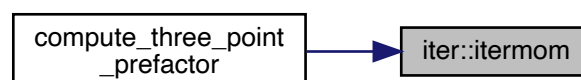
- `std::vector< Vector3d > itermom` (double `max_mom`, double `min_mom`, `ADAT::Array1dO< ADAT::Array1dO< int >> omit_mom`, bool `canonical`)

5.2.1 Function Documentation

5.2.1.1 `itermom()`

```
std::vector< Vector3d > iter::itermom (
    double max_mom,
    double min_mom,
    ADAT::Array1dO< ADAT::Array1dO< int >> omit_mom,
    bool canonical )
```

Here is the caller graph for this function:



5.3 KFactorEnv Namespace Reference

Functions

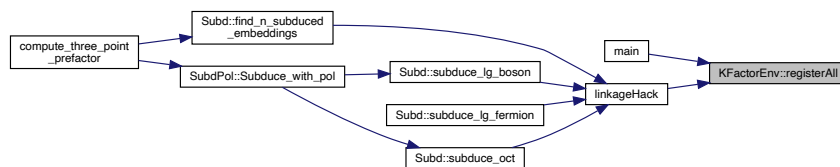
- bool [registerAll](#) ()

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

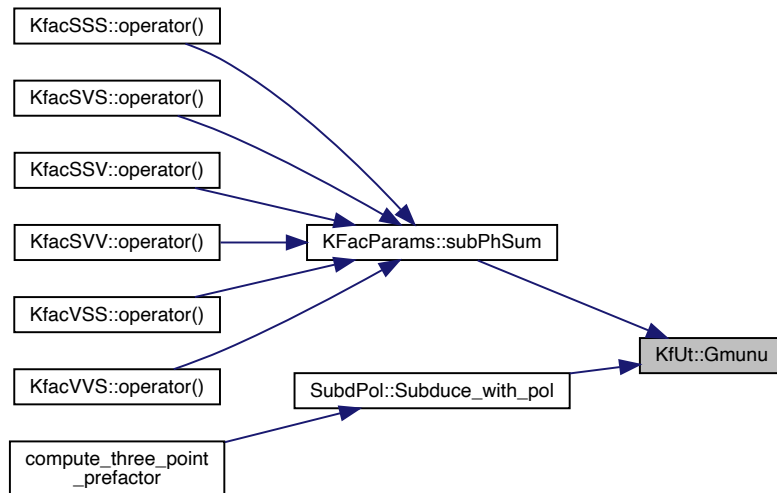
- double [truncate](#) (double num, int precision)
- Eigen::MatrixXcd [Gmunu](#) ()

5.4.1 Function Documentation

5.4.1.1 Gmunu()

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

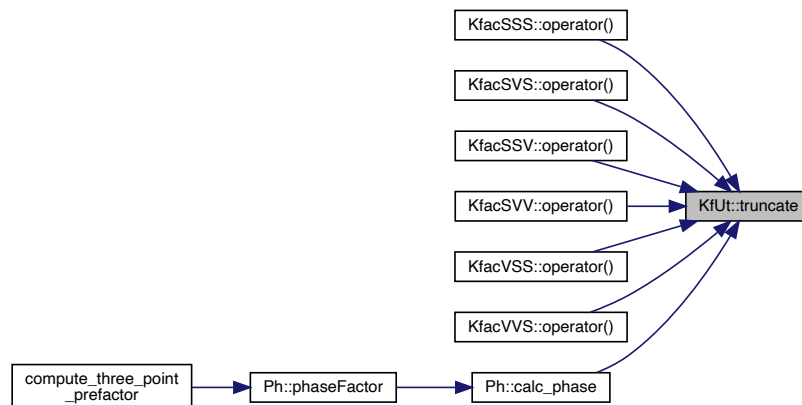
Here is the caller graph for this function:



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

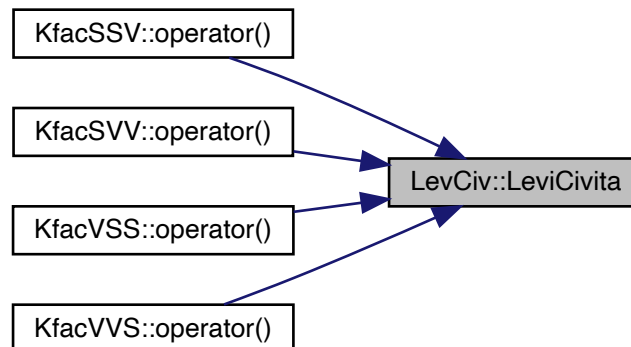
- double [LeviCivita](#) (int arr[], int n)

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_ )
```

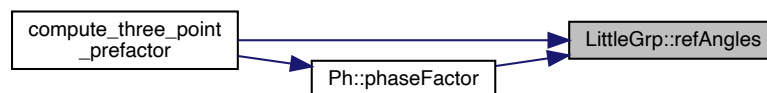
Here is the caller graph for this function:



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

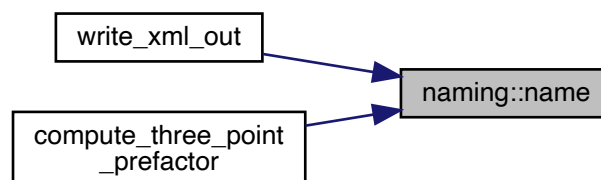
5.7.1.1 name()

```

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

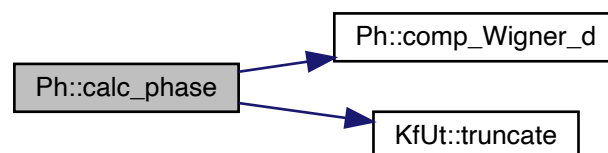
```
typedef std::tuple<int, int, int> Ph::tripKey
```

5.8.2 Function Documentation

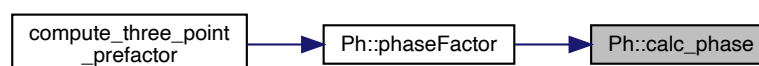
5.8.2.1 calc_phase()

```
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:



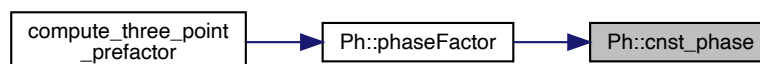
Here is the caller graph for this function:



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 )
```

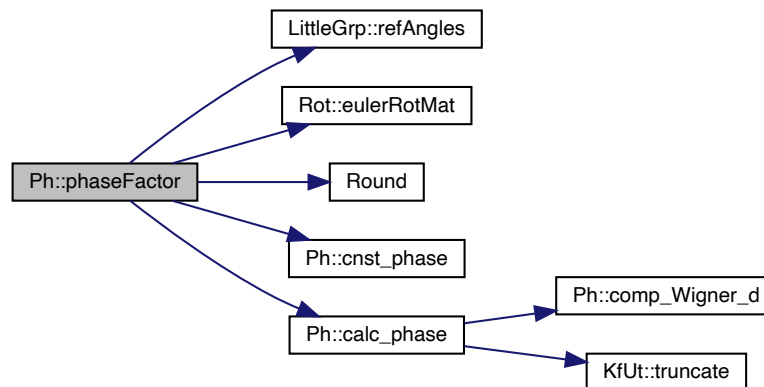
Here is the caller graph for this function:



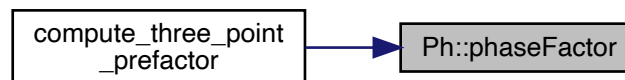
5.8.2.4 phaseFactor()

```
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

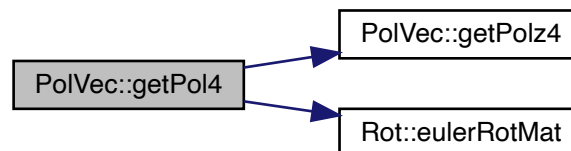
- Eigen::MatrixXcd [getPolz4](#) (double &mom_sq, const int &two_helicity, double &mass_sq, bool &curr)
- Eigen::MatrixXcd [getPol4](#) (double &mom_sq, const int &two_helicity, double &mass_sq, double &phi, double &theta, double &psi, bool curr)

5.9.1 Function Documentation

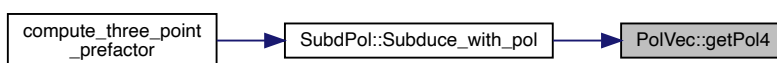
5.9.1.1 getPol4()

```
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:



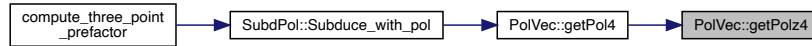
Here is the caller graph for this function:



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

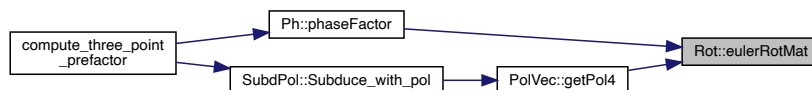
5.10.1.1 eulerRotMat()

```

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

```

Here is the caller graph for this function:



5.11 Subd Namespace Reference

Functions

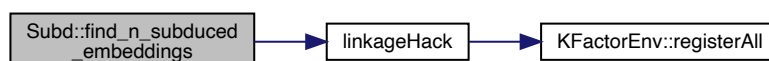
- map< int, complex< double > > [subduce_lg_boson](#) (const [irrep_label](#) &irrep, const string &little_group)
- map< int, complex< double > > [subduce_lg_fermion](#) (const [irrep_label](#) &irrep, const string &little_group)
- map< int, complex< double > > [subduce_oct](#) (const [irrep_label](#) &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:



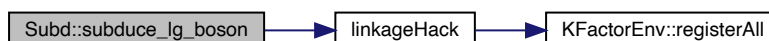
Here is the caller graph for this function:



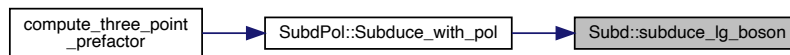
5.11.1.2 subduce_lg_boson()

```
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:

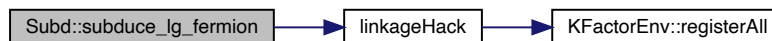


5.11.1.3 subduce_lg_fermion()

```

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

Here is the call graph for this function:

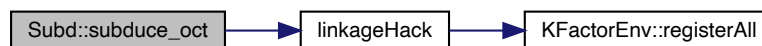


5.11.1.4 subduce_oct()

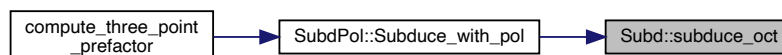
```

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

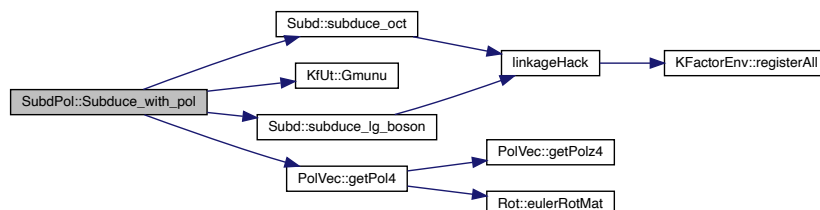
- `map< int, Eigen::MatrixXcd > Subduce_with_pol` (double &mom_sq, double &mass_sq, int &twoJ, const `irrep_label` &irrep, const string &little_group, double R1_phi, double R1_theta, double R1_psi, bool curr)

5.12.1 Function Documentation

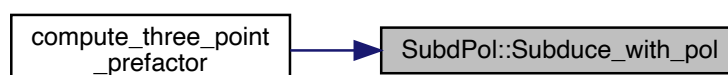
5.12.1.1 Subduce_with_pol()

```
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 flavour Struct Reference

```
#include <compute_kfactor_xml_read_write.h>
```

Data Fields

- int [threeY](#)
- int [twolz](#)
- int [twoI](#)

6.1.1 Field Documentation

6.1.1.1 threeY

```
int flavour::threeY
```

6.1.1.2 twoI

```
int flavour::twoI
```

6.1.1.3 twolz

```
int flavour::twoIz
```

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

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.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 IrrepLam_t Struct Reference

```
#include <compute_kfactor_xml_read_write.h>
```

Data Fields

- string [irrep](#)
- int [row](#)
- Eigen::Vector3d [mom](#)
- double [mom_sq](#)
- string [lev](#)
- int [two_lam](#)

6.3.1 Field Documentation

6.3.1.1 irrep

```
string IrrepLam_t::irrep
```

6.3.1.2 lev

```
string IrrepLam_t::lev
```

6.3.1.3 mom

```
Eigen::Vector3d IrrepLam_t::mom
```

6.3.1.4 mom_sq

```
double IrrepLam_t::mom_sq
```

6.3.1.5 row

```
int IrrepLam_t::row
```

6.3.1.6 two_lam

```
int IrrepLam_t::two_lam
```

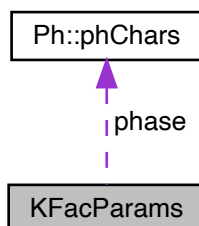
The documentation for this struct was generated from the following file:

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.h](#)

6.4 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](#)

6.4.1 Constructor & Destructor Documentation

6.4.1.1 ~KFacParams()

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

6.4.1.2 KFacParams()

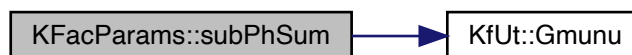
```
KFacParams::KFacParams (
    map< int, Eigen::MatrixXcd > Sub1_,
    map< int, Eigen::MatrixXcd > SubCurr_,
    map< int, Eigen::MatrixXcd > Sub3_,
    Ph::phChars phase_,
    VectorXd qp_,
    VectorXd qm_ )
```

6.4.2 Member Function Documentation

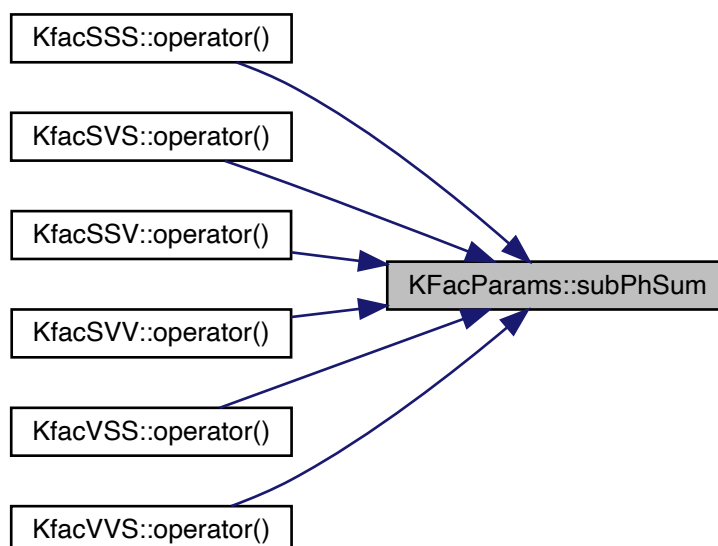
6.4.2.1 subPhSum()

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

Here is the call graph for this function:



Here is the caller graph for this function:



6.4.2.2 two_abs_lam()

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

6.4.3 Field Documentation

6.4.3.1 phase

`Ph::phChars KFacParams::phase`

6.4.3.2 qm

`VectorXd KFacParams::qm`

6.4.3.3 qp

`VectorXd KFacParams::qp`

6.4.3.4 Sub1

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

6.4.3.5 Sub3

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

6.4.3.6 SubCurr

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

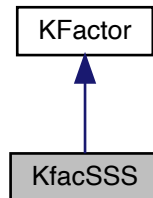
The documentation for this class was generated from the following files:

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

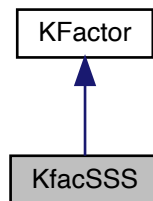
6.5 KfacSSS Class Reference

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

Inheritance diagram for KfacSSS:



Collaboration diagram for KfacSSS:



Public Member Functions

- `vector< complex< double > > operator() (const KFacParams ¶ms) const`
- `string name () const`

6.5.1 Member Function Documentation

6.5.1.1 name()

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

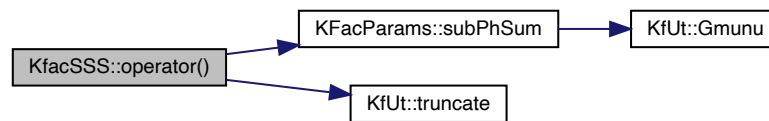
Implements [KFactor](#).

6.5.1.2 operator()

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

Implements [KFactor](#).

Here is the call graph for this function:



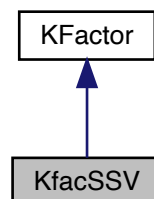
The documentation for this class was generated from the following files:

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h](#)
- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc](#)

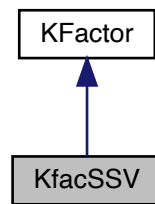
6.6 KfacSSV Class Reference

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

Inheritance diagram for KfacSSV:



Collaboration diagram for KfacSSV:



Public Member Functions

- `vector< complex< double > > operator() (const KFacParams ¶ms) const`
- `string name () const`

6.6.1 Member Function Documentation

6.6.1.1 name()

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

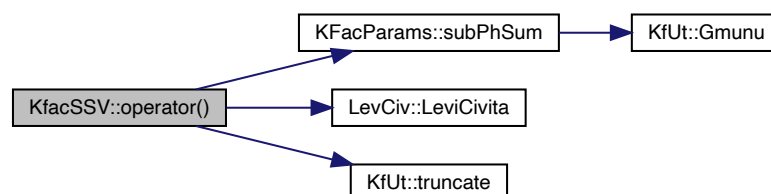
Implements [KFactor](#).

6.6.1.2 operator()

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

Implements [KFactor](#).

Here is the call graph for this function:



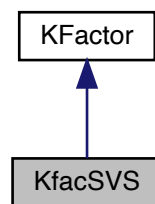
The documentation for this class was generated from the following files:

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h](#)
- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc](#)

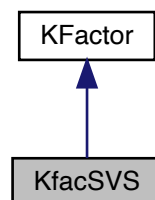
6.7 KfacSVS Class Reference

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

Inheritance diagram for KfacSVS:



Collaboration diagram for KfacSVS:



Public Member Functions

- `vector< complex< double > > operator() (const KFacParams ¶ms) const`
- `string name () const`

6.7.1 Member Function Documentation

6.7.1.1 name()

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

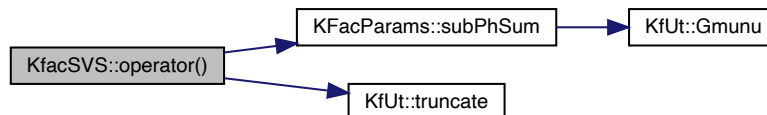
Implements [KFactor](#).

6.7.1.2 operator>()

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

Implements [KFactor](#).

Here is the call graph for this function:



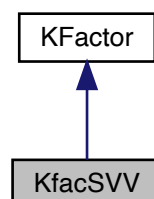
The documentation for this class was generated from the following files:

- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/[kfactor_pigammapi.h](#)
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/[kfactor_pigammapi.cc](#)

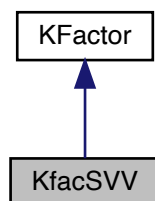
6.8 KfacSVV Class Reference

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

Inheritance diagram for KfacSVV:



Collaboration diagram for KfacSVV:



Public Member Functions

- `vector< complex< double > > operator() (const KFacParams ¶ms) const`
- `string name () const`

6.8.1 Member Function Documentation

6.8.1.1 name()

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

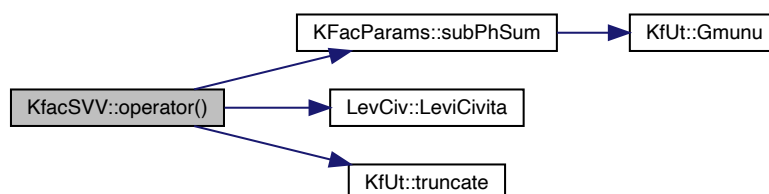
Implements [KFactor](#).

6.8.1.2 operator()

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

Implements [KFactor](#).

Here is the call graph for this function:



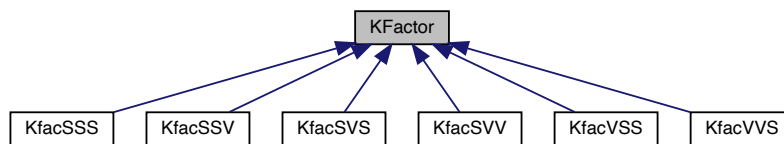
The documentation for this class was generated from the following files:

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

6.9 KFactor Class Reference

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

Inheritance diagram for KFactor:



Public Member Functions

- virtual [~KFactor](#) ()
- virtual `vector< complex< double > > operator()` (const [KFacParams](#) ¶ms) const =0
- virtual string [name](#) () const =0

6.9.1 Constructor & Destructor Documentation

6.9.1.1 ~KFactor()

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

6.9.2 Member Function Documentation

6.9.2.1 name()

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

Implemented in [KfacVSS](#), [KfacVVS](#), [KfacSSV](#), [KfacSVV](#), [KfacSSS](#), and [KfacSVS](#).

6.9.2.2 operator()

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

Implemented in [KfacVSS](#), [KfacVVS](#), [KfacSSV](#), [KfacSVV](#), [KfacSSS](#), and [KfacSVS](#).

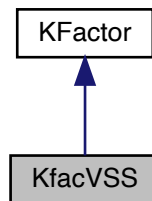
The documentation for this class was generated from the following file:

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h](#)

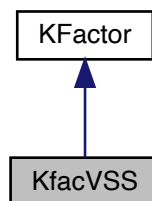
6.10 KfacVSS Class Reference

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

Inheritance diagram for KfacVSS:



Collaboration diagram for KfacVSS:



Public Member Functions

- `vector< complex< double > > operator() (const KFacParams ¶ms) const`
- `string name () const`

6.10.1 Member Function Documentation

6.10.1.1 name()

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

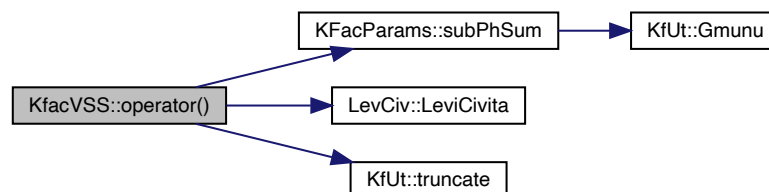
Implements [KFactor](#).

6.10.1.2 operator>()

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

Implements [KFactor](#).

Here is the call graph for this function:



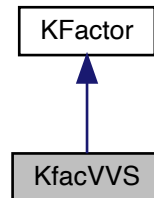
The documentation for this class was generated from the following files:

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h](#)
- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc](#)

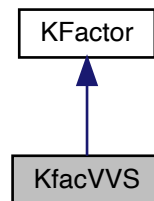
6.11 KfacVVS Class Reference

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

Inheritance diagram for KfacVVS:



Collaboration diagram for KfacVVS:



Public Member Functions

- `vector< complex< double > > operator() (const KFacParams ¶ms) const`
- `string name () const`

6.11.1 Member Function Documentation

6.11.1.1 name()

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

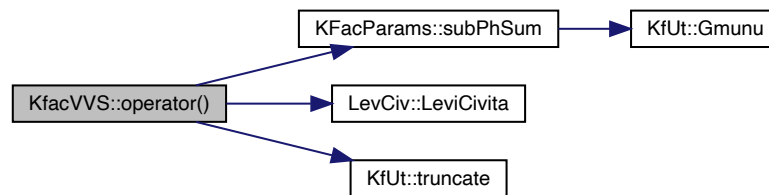
Implements [KFactor](#).

6.11.1.2 operator()

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

Implements [KFactor](#).

Here is the call graph for this function:



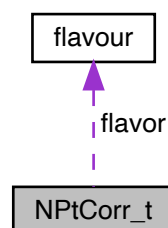
The documentation for this class was generated from the following files:

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h](#)
- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc](#)

6.12 NPtCorr_t Struct Reference

```
#include <compute_kfactor_xml_read_write.h>
```

Collaboration diagram for NPtCorr_t:



Data Fields

- string [name](#)
- Array1dO< string > [levels](#)
- int [twoJ](#)
- int [P](#)
- int [ell](#)
- ADAT::Array1dO< string > [elab](#)
- double [max_mom](#)
- double [min_mom](#)
- bool [canonical](#)
- ADAT::Array1dO< ADAT::Array1dO< int > > [omit_mom](#)
- [flavour](#) [flavor](#)
- bool [projected](#)
- bool [smearedP](#)
- int [t_slice](#)
- bool [creation_op](#)

6.12.1 Field Documentation

6.12.1.1 canonical

```
bool NPtCorr_t::canonical
```

6.12.1.2 creation_op

```
bool NPtCorr_t::creation_op
```

6.12.1.3 elab

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

6.12.1.4 ell

```
int NPtCorr_t::ell
```

6.12.1.5 flavor

```
flavour NPtCorr_t::flavor
```

6.12.1.6 levels

```
Array1dO<string> NPtCorr_t::levels
```

6.12.1.7 max_mom

```
double NPtCorr_t::max_mom
```

6.12.1.8 min_mom

```
double NPtCorr_t::min_mom
```

6.12.1.9 name

```
string NPtCorr_t::name
```

6.12.1.10 omit_mom

```
ADAT::Array1dO<ADAT::Array1dO<int> > NPtCorr_t::omit_mom
```

6.12.1.11 P

```
int NPtCorr_t::P
```

6.12.1.12 projected

```
bool NPtCorr_t::projected
```

6.12.1.13 smearedP

```
bool NPtCorr_t::smearedP
```

6.12.1.14 t_slice

```
int NPtCorr_t::t_slice
```

6.12.1.15 twoJ

```
int NPtCorr_t::twoJ
```

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

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.h](#)

6.13 NPtIrrepLam_t Struct Reference

```
#include <compute_kfactor_xml_read_write.h>
```

Data Fields

- [vector< IrrepLam_t > Npt](#)
- [Array1dO< Complex > kfac](#)

6.13.1 Field Documentation

6.13.1.1 kfac

```
Array1dO<Complex> NPtIrrepLam_t::kfac
```

6.13.1.2 Npt

```
vector<IrrepLam_t> NPtIrrepLam_t::Npt
```

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

- [/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.h](#)

6.14 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](#)

6.14.1 Member Function Documentation

6.14.1.1 [operator<\(\)](#)

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

6.14.2 Field Documentation

6.14.2.1 [lam_phase](#)

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

6.14.2.2 [mom1](#)

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

6.14.2.3 [mom2](#)

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

6.14.2.4 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](#)

6.15 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)

6.15.1 Member Function Documentation

6.15.1.1 toArray() [1/2]

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

Here is the caller graph for this function:



6.15.1.2 toArray() [2/2]

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

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

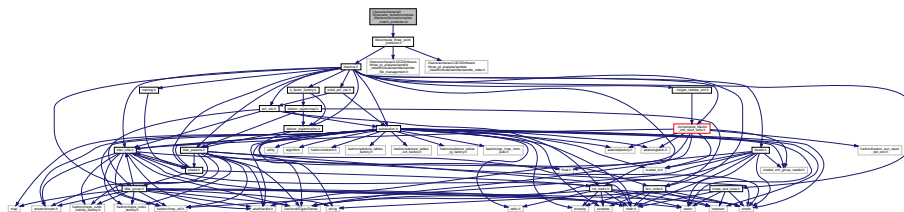
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/[kfac_utils.h](#)
- /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/[kfac_utils.cc](#)

Chapter 7

File Documentation

7.1 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_prefactor.cc File Reference

#include "lib/compute_three_point_prefactor.h"
Include dependency graph for compute_matrix_prefactor.cc:



Functions

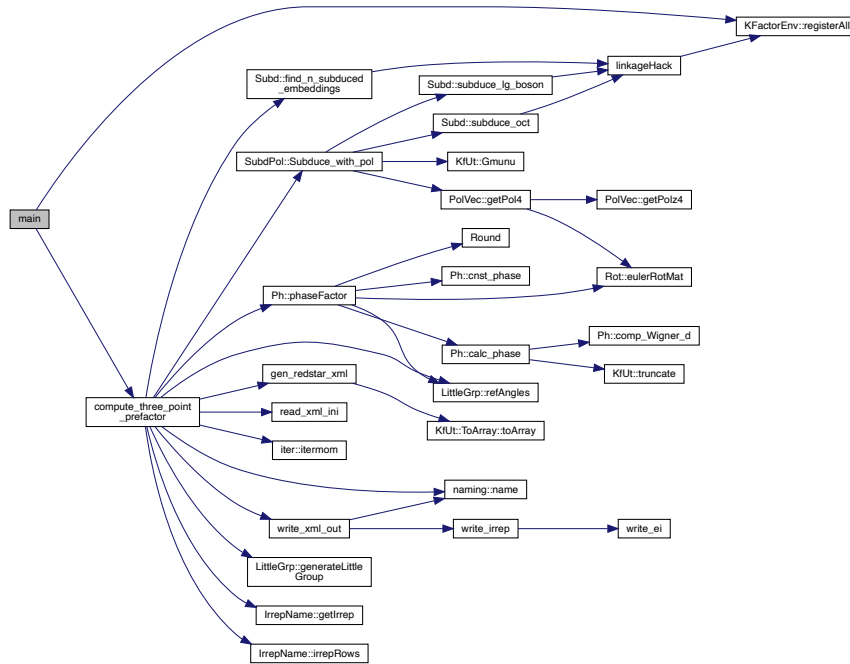
- int [main](#) (int argc, char **argv)

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/compute_kfactor_xml_read_write.cc File Reference

```
#include "compute_kfactor_xml_read_write.h"
```

Functions

- void [read_xml_ini](#) (XMLReader &xml_in, int &npt, int &L, double &Xi, double &XiE, std::vector< [NPtCorr_t](#) > &had, int &num_matrix_elem, string &matrix_type, Array1dO< string > &matrix_name, string &print_zero, string &compute_phase, string &elab_dir, bool &make_redstar_xml, string &redstar_xml)
- void [write_ei](#) (XMLWriter &xml, const std::string &path, const Eigen::Vector3d &input)
- void [write_irrep](#) (XMLWriter &xml_out, [IrrepLam_t](#) &irrep_lam)
- void [write_xml_out](#) (XMLWriter &xml_out, int &npt, int &L, double &Xi, double &XiE, std::vector< [NPtCorr_t](#) > &had, int &num_matrix_elem, string &matrix_type, Array1dO< string > &matrix_name, string &print_zero, string &compute_phase, string &elab_dir, bool &make_redstar_xml, vector< [NPtIrrepLam_t](#) > &irreps, int &count)

7.2.1 Function Documentation

7.2.1.1 read_xml_ini()

```
void read_xml_ini (
    XMLReader & xml_in,
    int & npt,
    int & L,
    double & Xi,
    double & XiE,
    std::vector< NPtCorr_t > & had,
    int & num_matrix_elem,
    string & matrix_type,
    ArrayldO< string > & matrix_name,
    string & print_zero,
    string & compute_phase,
    string & elab_dir,
    bool & make_redstar_xml,
    string & redstar_xml )
```

7.2.1.2 write_ei()

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

7.2.1.3 write_irrep()

```
void write_irrep (
    XMLWriter & xml_out,
    IrrepLam_t & irrep_lam )
```

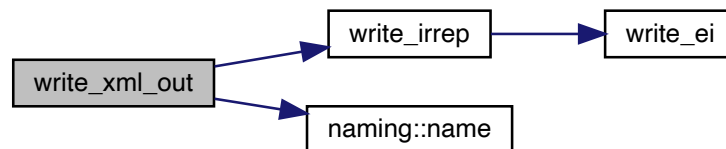
Here is the call graph for this function:



7.2.1.4 write_xml_out()

```
void write_xml_out (
    XMLWriter & xml_out,
    int & npt,
    int & L,
    double & Xi,
    double & XiE,
    std::vector< NPtCorr_t > & had,
    int & num_matrix_elem,
    string & matrix_type,
    Array1dO< string > & matrix_name,
    string & print_zero,
    string & compute_phase,
    string & elab_dir,
    bool & make_redstar_xml,
    vector< NPtIrrepLam_t > & irreps,
    int & count )
```

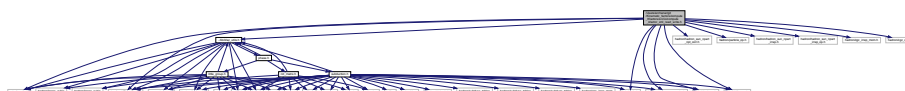
Here is the call graph for this function:



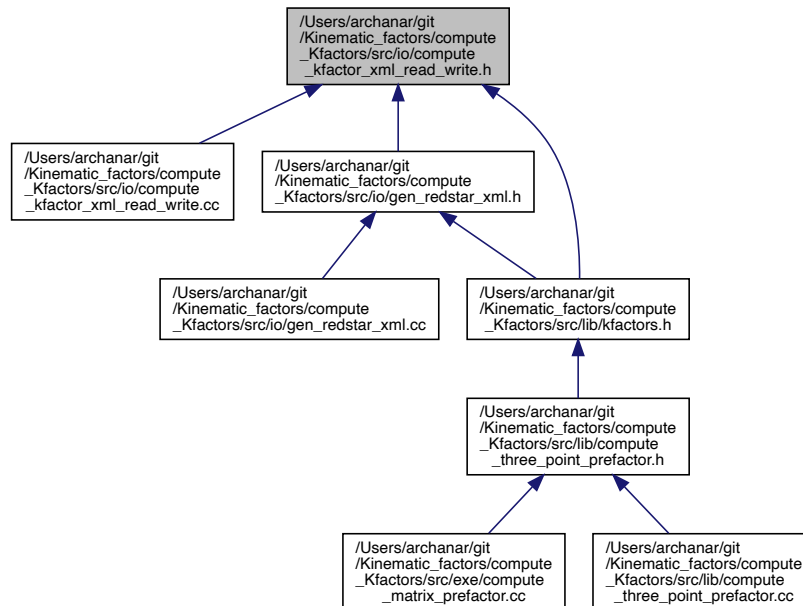
7.3 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_↔ xml_read_write.h File Reference

```
#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 `compute_kfactor_xml_read_write.h`:



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



Data Structures

- struct [IrrepLam_t](#)
- struct [NPtIrrepLam_t](#)
- struct [flavour](#)
- struct [NPtCorr_t](#)

Functions

- void [read_xml_ini](#) (XMLReader &xml_in, int &npt, int &L, double &Xi, double &XiE, std::vector< [NPtCorr_t](#) > &had, int &num_matrix_elem, string &matrix_type, Array1dO< string > &matrix_name, string &print_zero, string &compute_phase, string &elab_dir, bool &make_redstar_xml, string &redstar_xml)
- void [write_ei](#) (XMLWriter &xml, const std::string &path, const Eigen::Vector3d &input)
- void [write_irrep](#) (XMLWriter &xml_out, [IrrepLam_t](#) &irrep_lam)
- void [write_xml_out](#) (XMLWriter &xml_out, int &npt, int &L, double &Xi, double &XiE, std::vector< [NPtCorr_t](#) > &had, int &num_matrix_elem, string &matrix_type, Array1dO< string > &matrix_name, string &print_zero, string &compute_phase, string &elab_dir, bool &make_redstar_xml, vector< [NPtIrrepLam_t](#) > &irreps, int &count)

7.3.1 Function Documentation

7.3.1.1 read_xml_ini()

```
void read_xml_ini (
    XMLReader & xml_in,
    int & npt,
    int & L,
    double & Xi,
    double & XiE,
    std::vector< NPtCorr_t > & had,
    int & num_matrix_elem,
    string & matrix_type,
    ArrayldO< string > & matrix_name,
    string & print_zero,
    string & compute_phase,
    string & elab_dir,
    bool & make_redstar_xml,
    string & redstar_xml )
```

7.3.1.2 write_ei()

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

7.3.1.3 write_irrep()

```
void write_irrep (
    XMLWriter & xml_out,
    IrrepLam_t & irrep_lam )
```

Here is the call graph for this function:



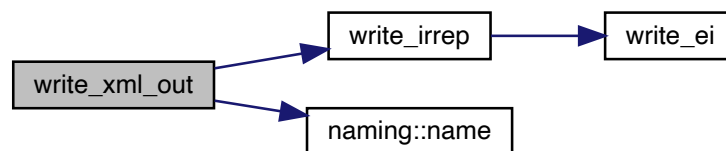
7.3.1.4 write_xml_out()

```

void write_xml_out (
    XMLWriter & xml_out,
    int & npt,
    int & L,
    double & Xi,
    double & XiE,
    std::vector< NPtCorr_t > & had,
    int & num_matrix_elem,
    string & matrix_type,
    Array1d0< string > & matrix_name,
    string & print_zero,
    string & compute_phase,
    string & elab_dir,
    bool & make_redstar_xml,
    vector< NPtIrrepLam_t > & irreps,
    int & count )

```

Here is the call graph for this function:

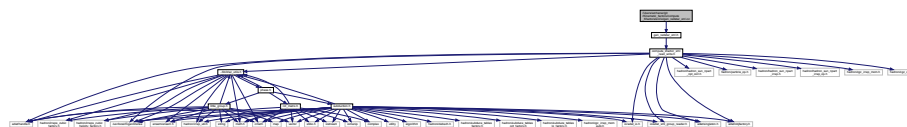


7.4 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.cc File Reference

```

#include "gen_redstar_xml.h"
Include dependency graph for gen_redstar_xml.cc:

```



Functions

- void `gen_redstar_xml` (vector< NPtCorr_t > &had, vector< NPtIrrepLam_t > &irreps, XMLWriter &red_xml)

7.4.1 Function Documentation

7.4.1.1 gen_redstar_xml()

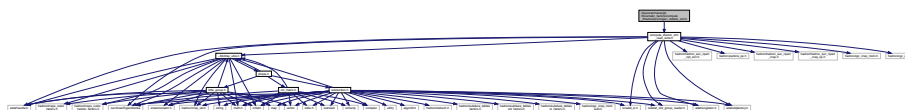
```
void gen_redstar_xml (
    vector< NPtCorr_t > & had,
    vector< NPtIrrepLam_t > & irreps,
    XMLWriter & red_xml )
```

Here is the call graph for this function:

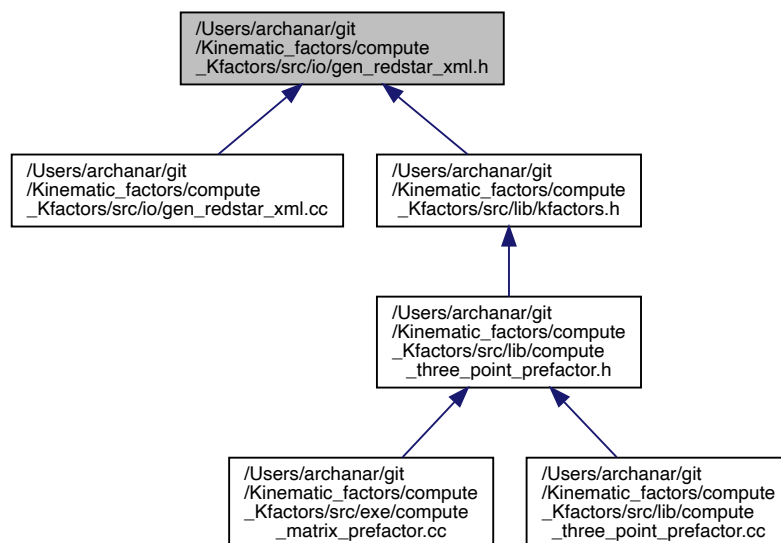


7.5 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_redstar_xml.h File Reference

```
#include "compute_kfactor_xml_read_write.h"
Include dependency graph for gen_redstar_xml.h:
```



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



Functions

- void [gen_redstar_xml](#) (vector< [NPtCorr_t](#) > &had, vector< [NPtIrrepLam_t](#) > &irreps, XMLWriter &red_xml)

7.5.1 Function Documentation

7.5.1.1 [gen_redstar_xml\(\)](#)

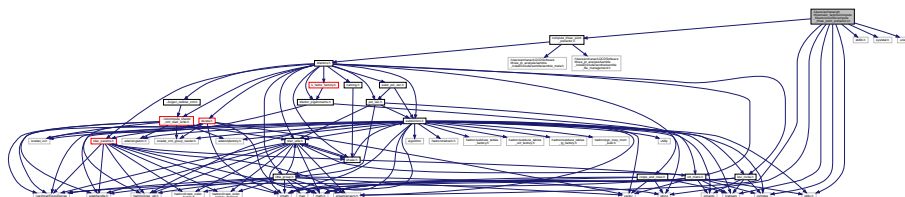
```
void gen_redstar_xml (
    vector< NPtCorr\_t > & had,
    vector< NPtIrrepLam\_t > & irreps,
    XMLWriter & red_xml )
```

Here is the call graph for this function:



7.6 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.cc File Reference ↔

```
#include "compute_three_point_prefactor.h"
Include dependency graph for compute_three_point_prefactor.cc:
```



Functions

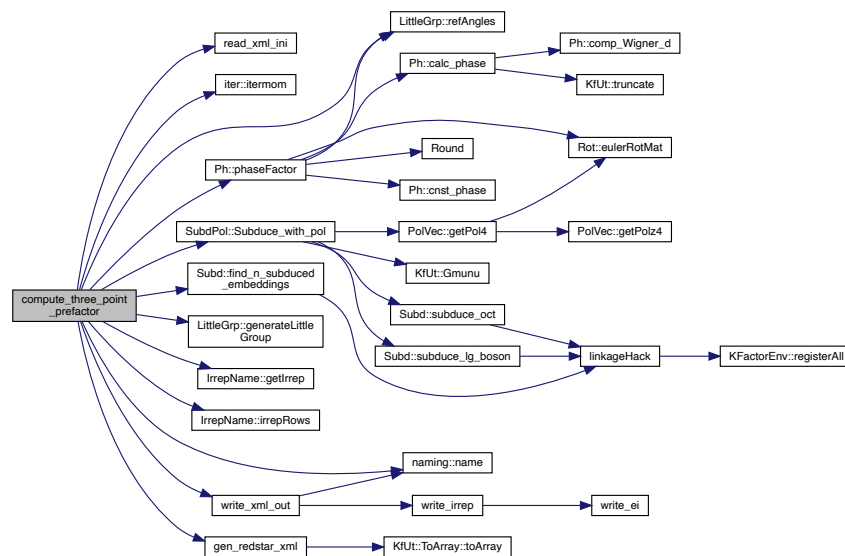
- void [compute_three_point_prefactor](#) (string &in, string &out)

7.6.1 Function Documentation

7.6.1.1 compute_three_point_prefactor()

```
void compute_three_point_prefactor (
    string & in,
    string & out )
```

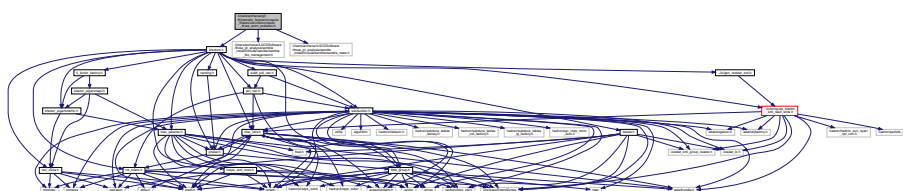
Here is the call graph for this function:



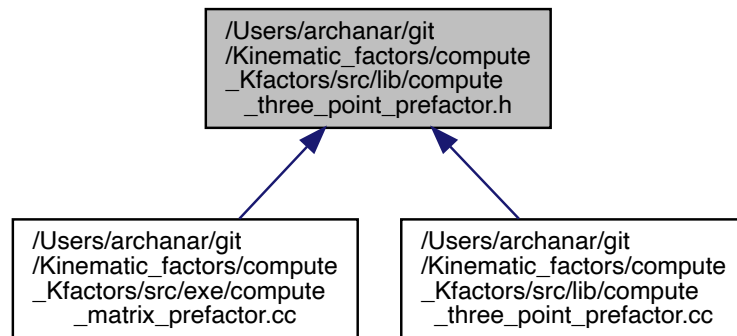
7.7 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.h File Reference

```
#include "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_three_point_prefactor.h:



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



Functions

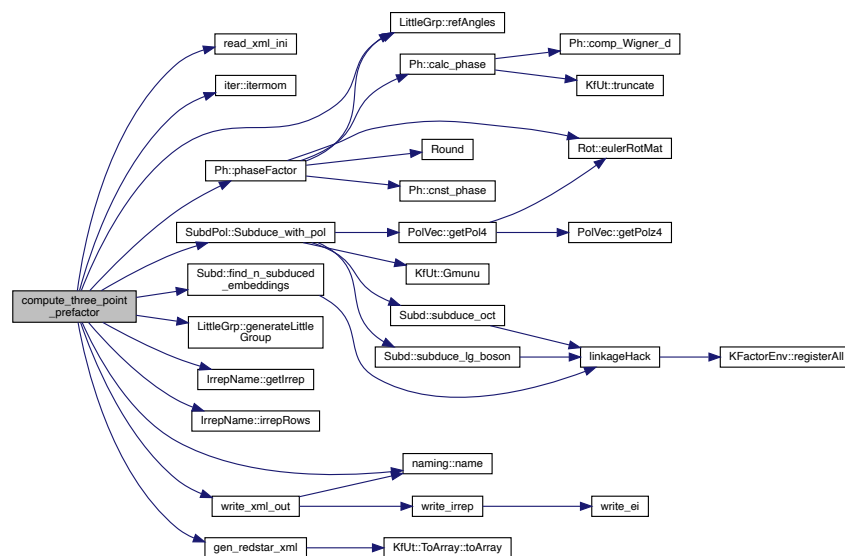
- void `compute_three_point_prefactor` (string &in, string &out)

7.7.1 Function Documentation

7.7.1.1 `compute_three_point_prefactor()`

```
void compute_three_point_prefactor (  
    string & in,  
    string & out )
```

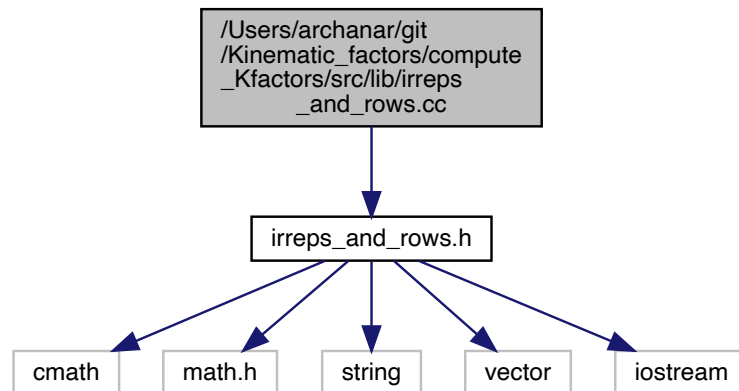
Here is the call graph for this function:



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

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

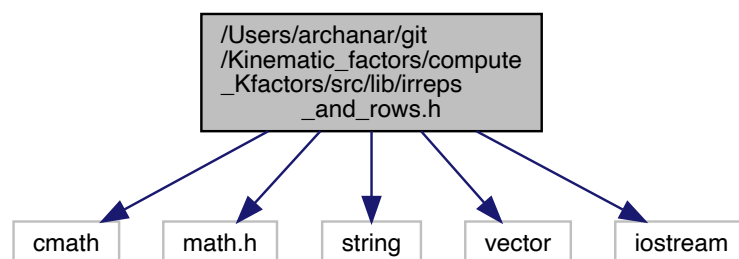
Include dependency graph for irreps_and_rows.cc:



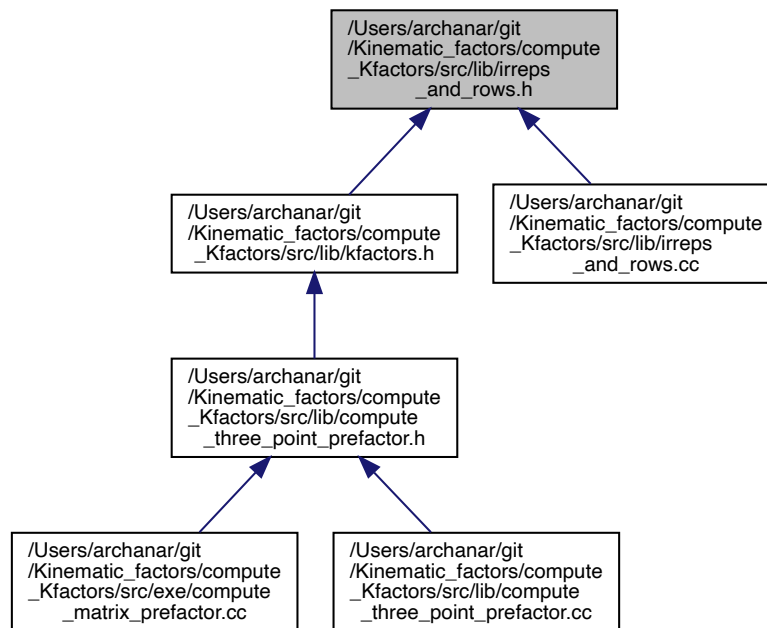
7.9 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h File Reference

```
#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](#)

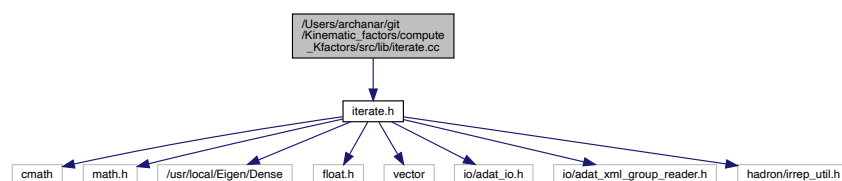
Functions

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

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

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

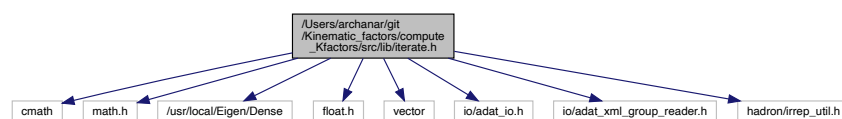
Include dependency graph for `iterate.cc`:



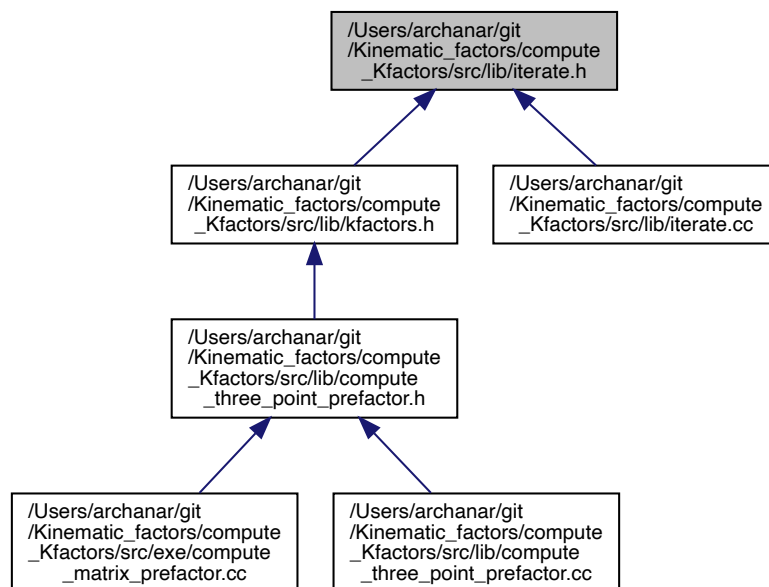
7.11 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h File Reference

```
#include <cmath>
#include "math.h"
#include </usr/local/Eigen/Dense>
#include <float.h>
#include <vector>
#include "io/adat_io.h"
#include "io/adat_xml_group_reader.h"
#include "hadron/irrep_util.h"
```

Include dependency graph for iterate.h:



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



Namespaces

- [iter](#)

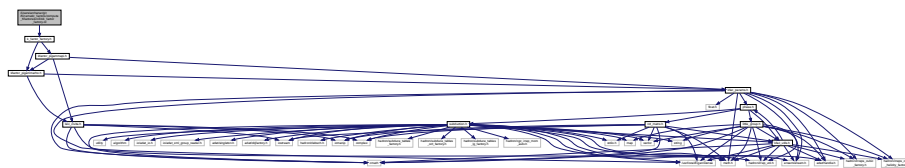
Functions

- `std::vector< Vector3d > iter::itermom` (double max_mom, double min_mom, ADAT::Array1dO< ADAT::Array1dO< int >> omit_mom, bool canonical)

7.12 /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

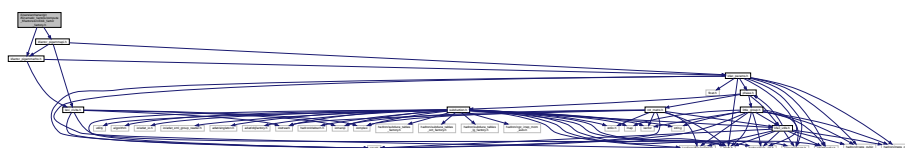
- `bool KFactorEnv::registerAll ()`

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

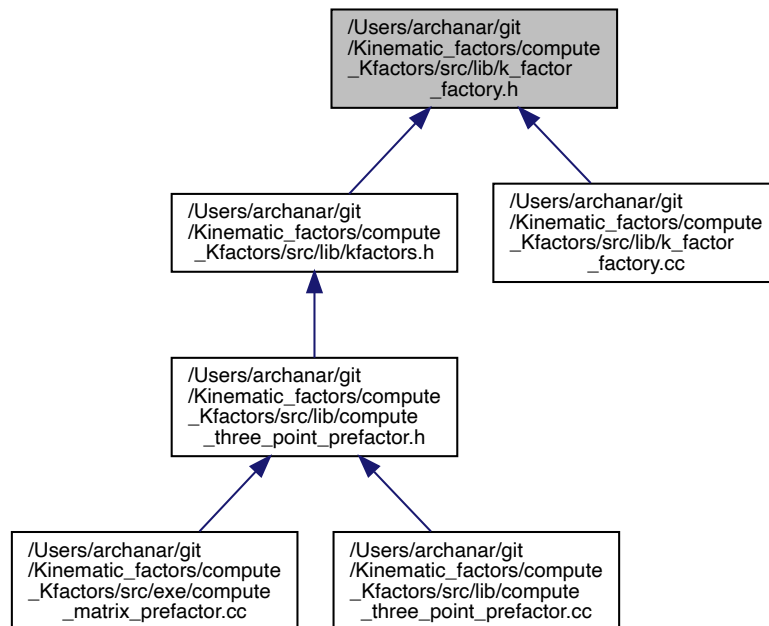
```
#include "kfactor_pigammarho.h"
```

```
#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](#) ()

7.13.1 Typedef Documentation

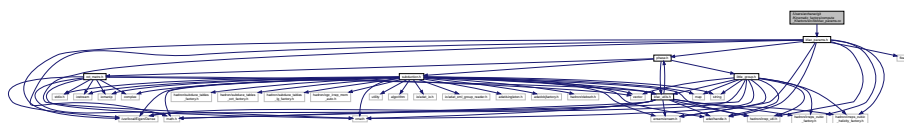
7.13.1.1 TheKFactorFactory

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


7.14 /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:



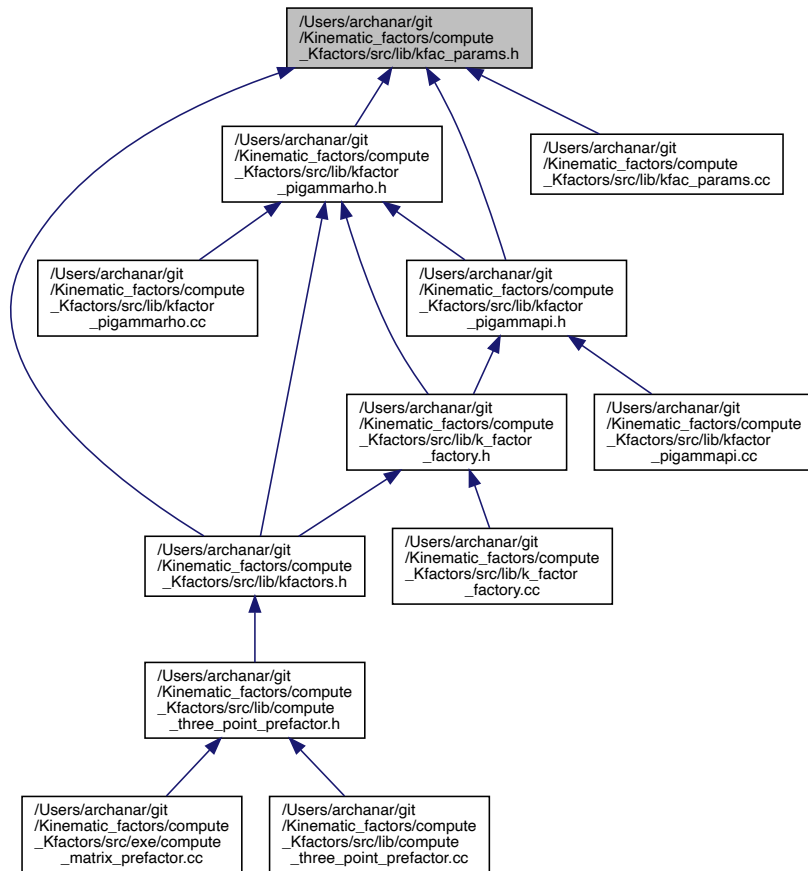
7.15 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.h File Reference

```
#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:



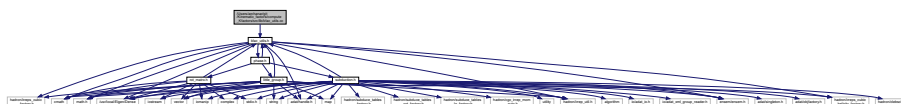
Data Structures

- class [KFacParams](#)

7.16 /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

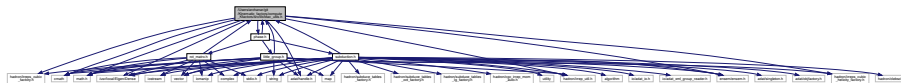
- [KfUt](#)

Functions

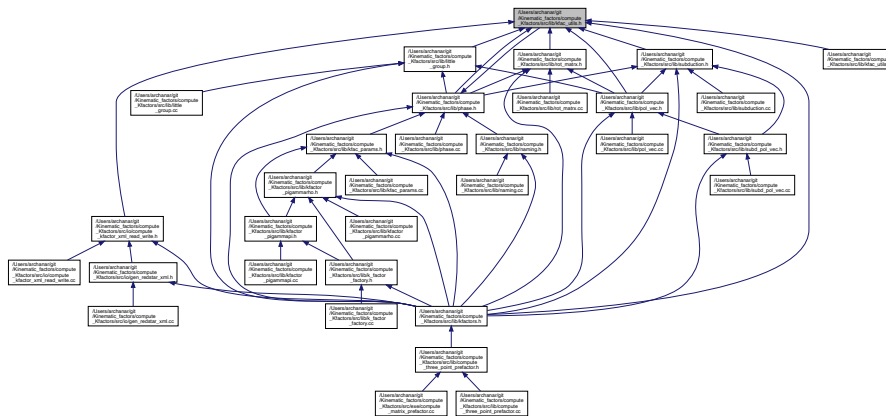
- double [KfUt::truncate](#) (double num, int precision)
- Eigen::MatrixXcd [KfUt::Gmunu](#) ()

7.17 /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

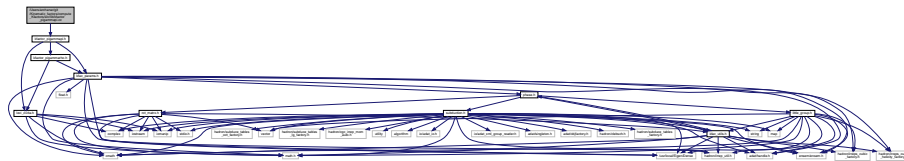
- [KfUt](#)

Functions

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

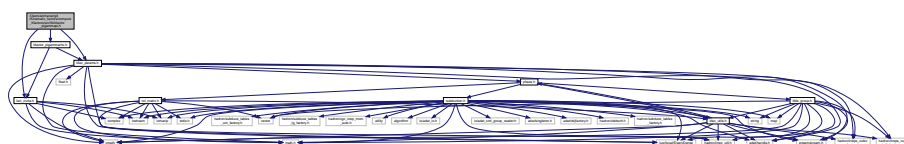
7.18 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.cc File Reference

```
#include "kfactor_pigammapi.h"
Include dependency graph for kfactor_pigammapi.cc:
```

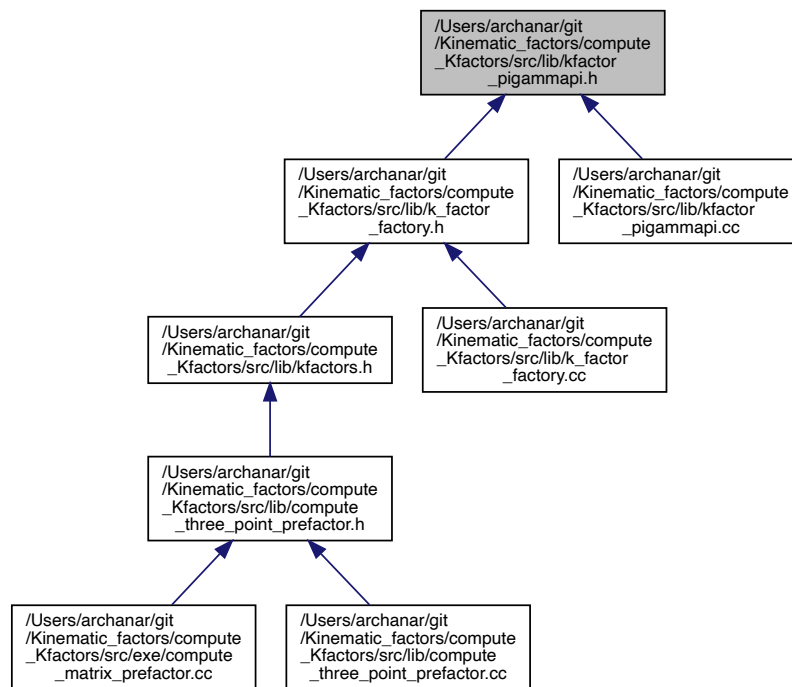


7.19 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammapi.h File Reference

```
#include "kfacs_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:



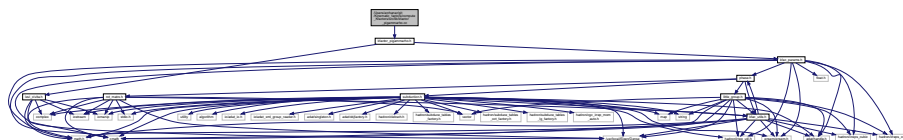
Data Structures

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

7.20 /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`:

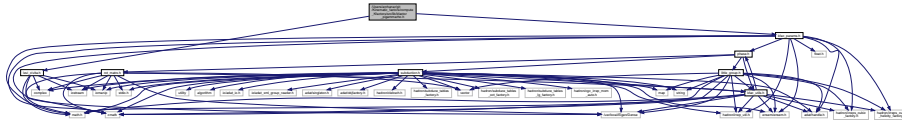


7.21 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h File Reference

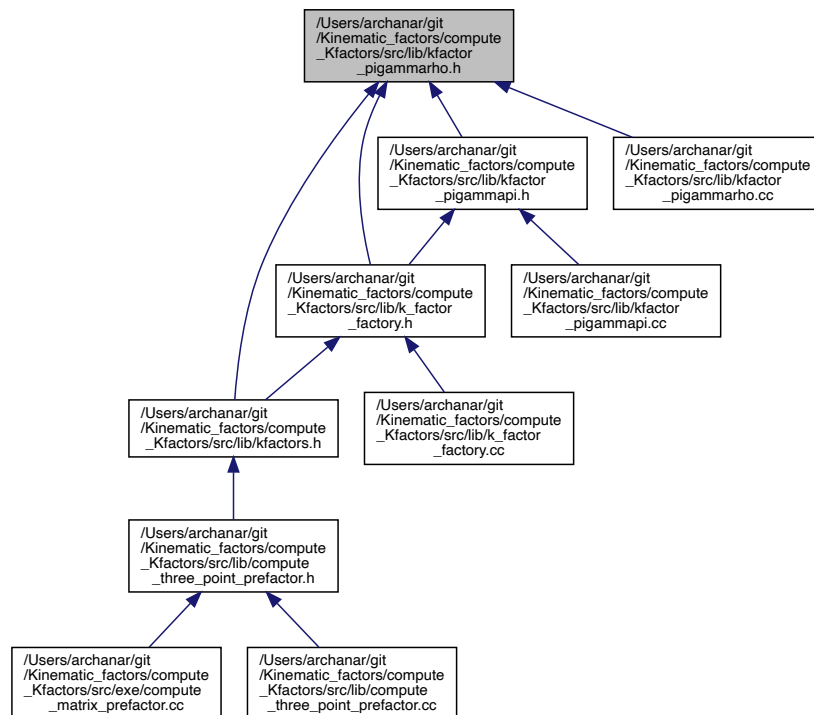
```
#include "kfac_params.h"
```

```
#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](#)

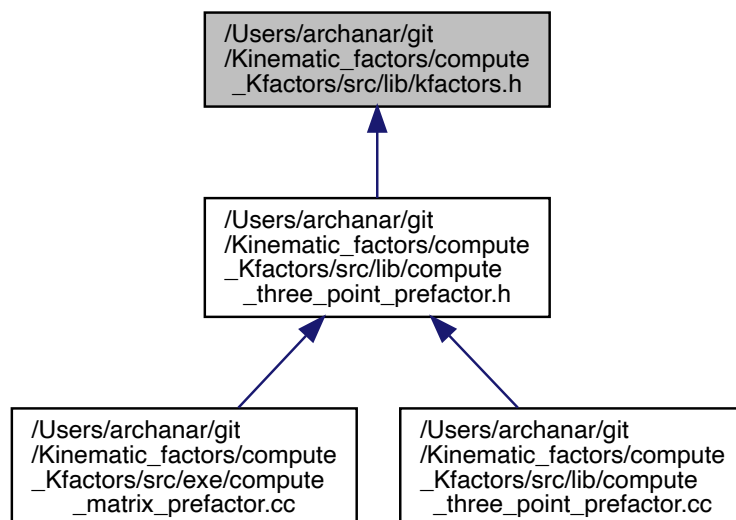
7.22 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactors.h File Reference

```
#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 "k_factor_factory.h"
#include "kfac_params.h"
#include "kfactor_pigammarho.h"
#include "naming.h"
#include "iterate.h"
#include "../io/compute_kfactor_xml_read_write.h"
#include "../io/gen_redstar_xml.h"
```

Include dependency graph for kfactors.h:



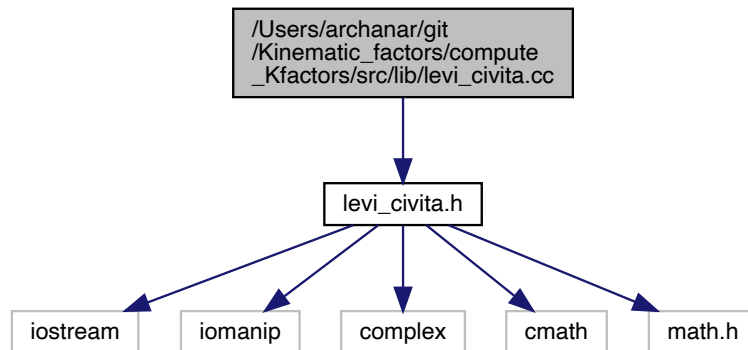
This graph shows which files directly or indirectly include this file:



7.23 /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:



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

```
#include <iostream>
```

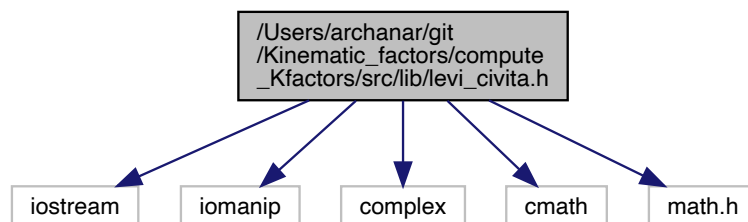
```
#include <iomanip>
```

```
#include <complex>
```

```
#include <cmath>
```

```
#include "math.h"
```

Include dependency graph for levi_civita.h:




```

graph TD
    A["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/levi_civita.h"]
    B["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/kfactor  
_pigammarho.h"]
    C["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/levi_civita.cc"]
    D["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/kfactor  
_pigammarho.cc"]
    E["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/kfactor  
_pigammmapi.h"]
    F["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/kfactor  
_pigammmapi.cc"]
    G["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/k_factor  
_factory.h"]
    H["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/k_factor  
_factory.cc"]
    I["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/kfactors.h"]
    J["/Users/archanar/git  
/Kinematic_factors/compute  
_Kfactors/src/lib/compute  
_three_point_prefactor.h"]
    K["/Users/archanar/git  
/Kinematic_factors/src/exe/compute  
_matrix_prefactor.cc"]
    L["/Users/archanar/git  
/Kinematic_factors/src/lib/compute  
_three_point_prefactor.cc"]

    A --> B
    A --> C
    B --> D
    B --> E
    E --> F
    D --> G
    G --> H
    G --> I
    I --> J
    I --> L
    J --> K
    J --> L
  
```

- LevCiv

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

```
#include "little_group.h"
Include dependency graph for little_group.cc:
```



Namespaces

- [LittleGrp](#)

Functions

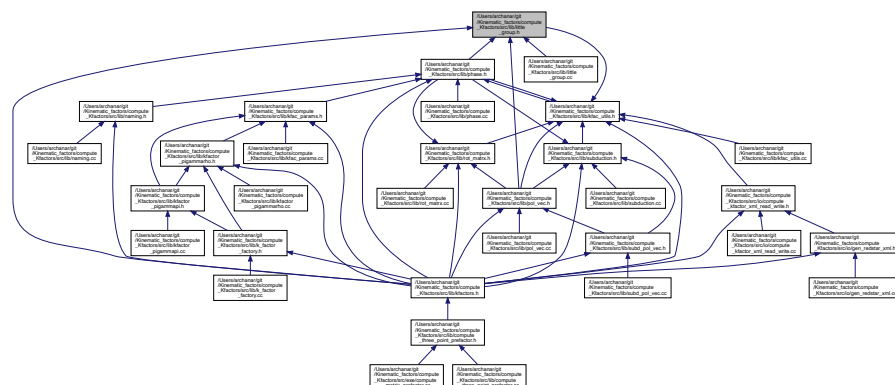
- string [LittleGrp::generateLittleGroup](#) (Eigen::Vector3d &mom_)
- std::vector< double > [LittleGrp::refAngles](#) (Eigen::Vector3d mom1)

7.26 /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)

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

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

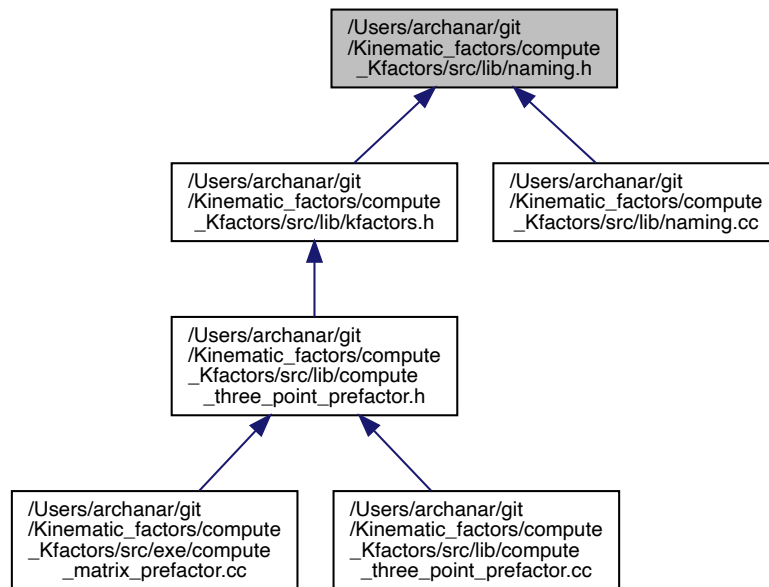


7.28 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h File Reference

```
#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)

7.29 /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)

7.29.1 Function Documentation

7.29.1.1 Round()

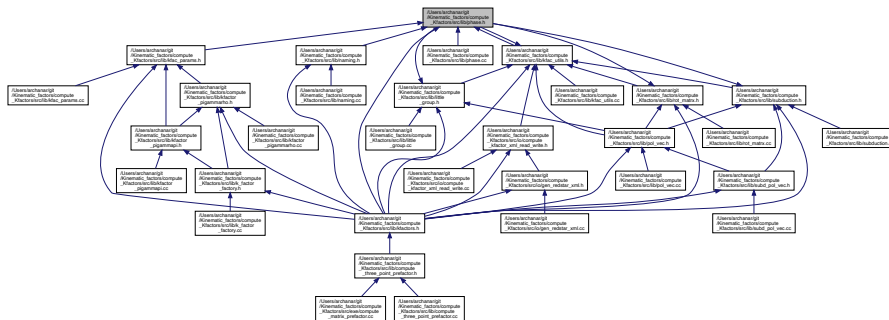
```
double Round (
    double x )
```

7.30 /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)

7.30.1 Function Documentation

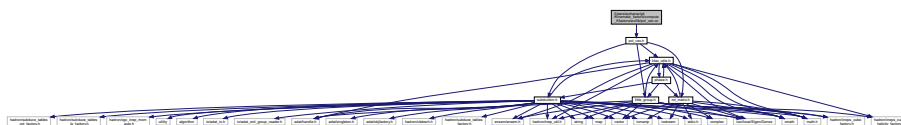
7.30.1.1 Round()

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

7.31 /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

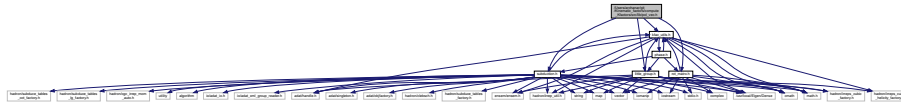
- [PolVec](#)

Functions

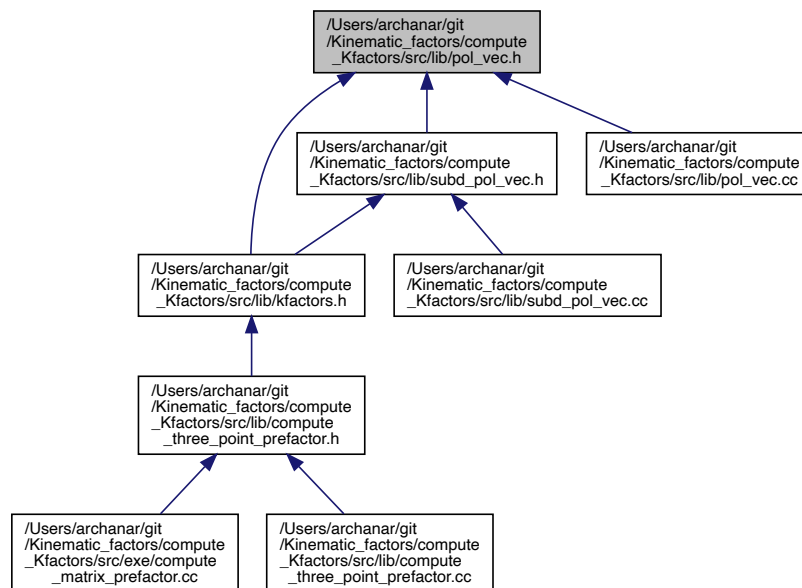
- Eigen::MatrixXcd [PolVec::getPolz4](#) (double &mom_sq, const int &two_helicity, double &mass_sq, bool &curr)
- Eigen::MatrixXcd [PolVec::getPol4](#) (double &mom_sq, const int &two_helicity, double &mass_sq, double &phi, double &theta, double &psi, bool curr)

7.32 /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:



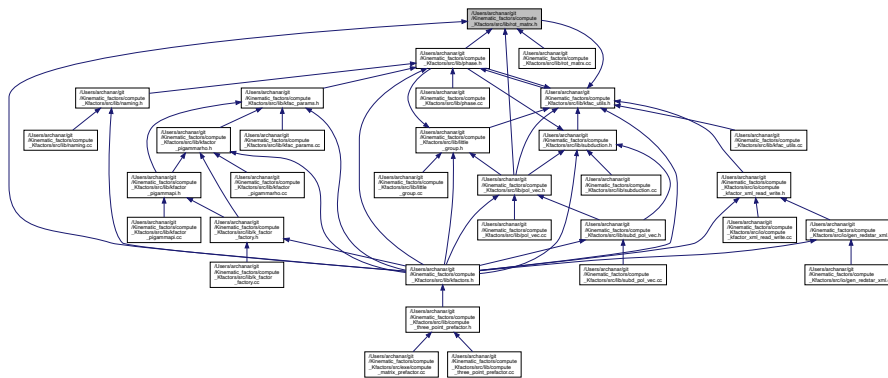
Namespaces

- [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)

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



Namespaces

- [Rot](#)

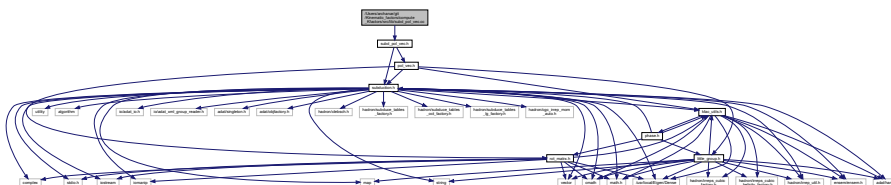
Functions

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

7.35 /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`:

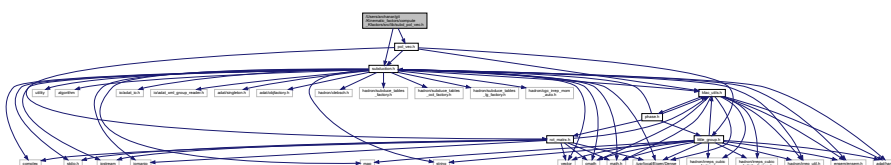


7.36 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol_vec.h File Reference

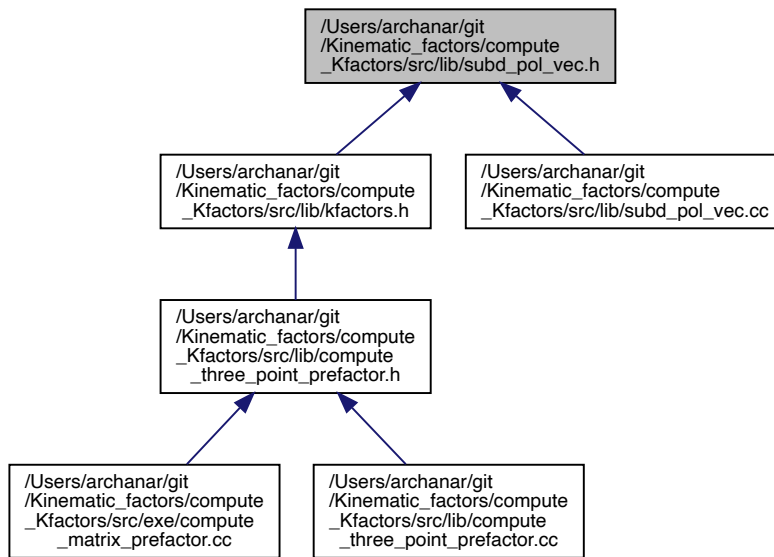
```
#include "subduction.h"
```

```
#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](#)

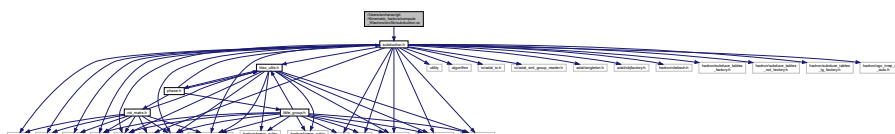
Functions

- `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)`

7.37 /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)`

7.37.1 Function Documentation

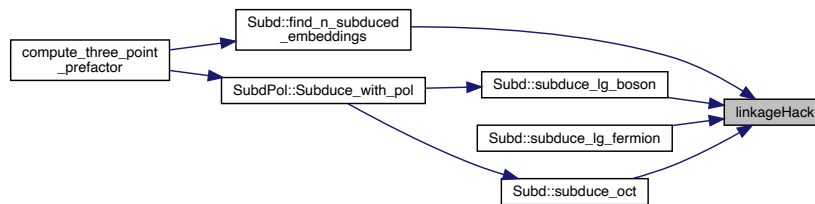
7.37.1.1 linkageHack()

```
bool linkageHack (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:

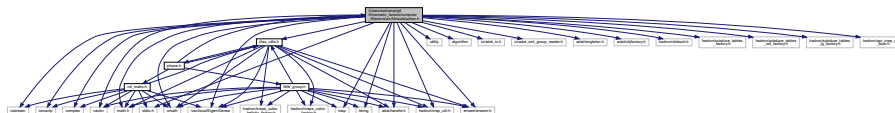


7.38 /Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h File Reference

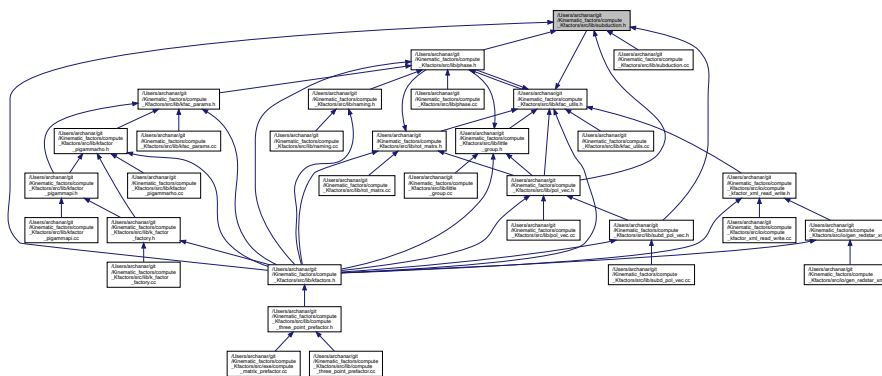
```
#include <vector>
#include <iostream>
#include <iomanip>
#include <map>
#include <string>
#include <complex>
#include <utility>
#include <algorithm>
#include <cmath>
#include "math.h"
#include <stdio.h>
#include </usr/local/Eigen/Dense>
#include "io/adat_io.h"
#include "io/adat_xml_group_reader.h"
```

```
#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

/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/exe/compute_matrix_prefactor.cc, 49
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/little_group.cc, 49
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.cc, 50
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.cc, 50
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/compute_kfactor_xml_read_write.h, 52
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/naming.h, 52
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_reestar_xml.cc, 55
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.cc, 55
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/io/gen_reestar_xml.h, 56
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/phase.h, 56
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.cc, 57
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.cc, 57
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/compute_three_point_prefactor.h, 58
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/pol_vec.h, 58
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.cc, 60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrix.cc, 60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/irreps_and_rows.h, 60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/rot_matrix.h, 60
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.cc, 61
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol.cc, 61
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/iterate.h, 62
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subd_pol.h, 62
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.cc, 63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.cc, 63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/k_factor_factory.h, 63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/subduction.h, 63
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_params.cc, 65
~KFacParams
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/KFacParams.h, 65
~KFactor
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/KFactor.h, 65
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.cc, 66
calc_phase
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfac_utils.h, 67
ph, 48
canonical
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigamapi.cc, 68
NPICOF, 43
cnst_phase
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigamapi.h, 68
ph, 19
comp_Wigner_d
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc, 69
ph, 19
compute_kfactor_xml_read_write.cc
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h, 70
read_xml, 50
write_ei, 51
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc, 71
write_xml_out, 51
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h, 72
read_xml_ini, 53
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.cc, 72
write_irrep, 54
/Users/archanar/git/Kinematic_factors/compute_Kfactors/src/lib/kfactor_pigammarho.h, 72
little_group.cc, 54

- compute_matrix_prefactor.cc
 - main, [49](#)
- compute_three_point_prefactor
 - compute_three_point_prefactor.cc, [57](#)
 - compute_three_point_prefactor.h, [59](#)
- compute_three_point_prefactor.cc
 - compute_three_point_prefactor, [57](#)
- compute_three_point_prefactor.h
 - compute_three_point_prefactor, [59](#)
- creation_op
 - NPtCorr_t, [43](#)
- elab
 - NPtCorr_t, [43](#)
- ell
 - NPtCorr_t, [43](#)
- eulerRotMat
 - Rot, [20](#)
- find_n_subduced_embeddings
 - Subd, [20](#)
- flavor
 - NPtCorr_t, [43](#)
- flavour, [25](#)
 - threeY, [25](#)
 - twol, [25](#)
 - twolz, [25](#)
- gen_redstar_xml
 - gen_redstar_xml.cc, [55](#)
 - gen_redstar_xml.h, [57](#)
- gen_redstar_xml.cc
 - gen_redstar_xml, [55](#)
- gen_redstar_xml.h
 - gen_redstar_xml, [57](#)
- generateLittleGroup
 - LittleGrp, [13](#)
- getIrrep
 - IrrepName, [9](#)
- getPol4
 - PolVec, [19](#)
- getPolz4
 - PolVec, [19](#)
- Gmunu
 - KfUt, [11](#)
- irrep
 - irrep_label, [26](#)
 - IrrepLam_t, [27](#)
- irrep_label, [26](#)
 - irrep, [26](#)
 - n, [26](#)
 - operator<, [26](#)
 - P, [26](#)
 - row, [27](#)
 - twoJ, [27](#)
- IrrepLam_t, [27](#)
 - irrep, [27](#)
 - lev, [27](#)
 - mom, [28](#)
 - mom_sq, [28](#)
 - row, [28](#)
 - two_lam, [28](#)
- IrrepName, [9](#)
 - getIrrep, [9](#)
 - irrepRows, [9](#)
- irrepRows
 - IrrepName, [9](#)
- iter, [10](#)
 - itermom, [10](#)
- itermom
 - iter, [10](#)
- k_factor_factory.h
 - TheKFactorFactory, [64](#)
- kfac
 - NPtIrrepLam_t, [45](#)
- KFacParams, [28](#)
 - ~KFacParams, [29](#)
 - KFacParams, [29](#)
 - phase, [30](#)
 - qm, [31](#)
 - qp, [31](#)
 - Sub1, [31](#)
 - Sub3, [31](#)
 - SubCurr, [31](#)
 - subPhSum, [29](#)
 - two_abs_lam, [30](#)
- KfacSSS, [32](#)
 - name, [32](#)
 - operator(), [32](#)
- KfacSSV, [33](#)
 - name, [34](#)
 - operator(), [34](#)
- KfacSVS, [35](#)
 - name, [35](#)
 - operator(), [36](#)
- KfacSVV, [36](#)
 - name, [37](#)
 - operator(), [37](#)
- KFactor, [38](#)
 - ~KFactor, [38](#)
 - name, [38](#)
 - operator(), [38](#)
- KFactorEnv, [11](#)
 - registerAll, [11](#)
- KfacVSS, [39](#)
 - name, [40](#)
 - operator(), [40](#)
- KfacVVS, [41](#)
 - name, [41](#)
 - operator(), [41](#)
- KfUt, [11](#)
 - Gmunu, [11](#)
 - truncate, [12](#)
- KfUt::ToArray, [47](#)
 - toArray, [47](#)

- lam_phase
 - Ph::phChars, [46](#)
- lev
 - IrrepLam_t, [27](#)
- LevCiv, [13](#)
 - LeviCivita, [13](#)
- levels
 - NPtCorr_t, [44](#)
- LeviCivita
 - LevCiv, [13](#)
- linkageHack
 - subduction.cc, [83](#)
- LittleGrp, [13](#)
 - generateLittleGroup, [13](#)
 - refAngles, [14](#)
- main
 - compute_matrix_prefactor.cc, [49](#)
- max_mom
 - NPtCorr_t, [44](#)
- min_mom
 - NPtCorr_t, [44](#)
- mom
 - IrrepLam_t, [28](#)
- mom1
 - Ph::phChars, [46](#)
- mom2
 - Ph::phChars, [46](#)
- mom_sq
 - IrrepLam_t, [28](#)
- n
 - irrep_label, [26](#)
- name
 - KfacSSS, [32](#)
 - KfacSSV, [34](#)
 - KfacSVS, [35](#)
 - KfacSVV, [37](#)
 - KFactor, [38](#)
 - KfacVSS, [40](#)
 - KfacVVS, [41](#)
 - naming, [14](#)
 - NPtCorr_t, [44](#)
- naming, [14](#)
 - name, [14](#)
- Npt
 - NPtIrrepLam_t, [45](#)
- NPtCorr_t, [42](#)
 - canonical, [43](#)
 - creation_op, [43](#)
 - elab, [43](#)
 - ell, [43](#)
 - flavor, [43](#)
 - levels, [44](#)
 - max_mom, [44](#)
 - min_mom, [44](#)
 - name, [44](#)
 - omit_mom, [44](#)
 - P, [44](#)
 - projected, [44](#)
 - smearedP, [44](#)
 - t_slice, [45](#)
 - twoJ, [45](#)
- NPtIrrepLam_t, [45](#)
 - kfac, [45](#)
 - Npt, [45](#)
- omit_mom
 - NPtCorr_t, [44](#)
- operator<
 - irrep_label, [26](#)
 - Ph::phChars, [46](#)
- operator()
 - KfacSSS, [32](#)
 - KfacSSV, [34](#)
 - KfacSVS, [36](#)
 - KfacSVV, [37](#)
 - KFactor, [38](#)
 - KfacVSS, [40](#)
 - KfacVVS, [41](#)
- P
 - irrep_label, [26](#)
 - NPtCorr_t, [44](#)
- Ph, [15](#)
 - calc_phase, [16](#)
 - cnst_phase, [16](#)
 - comp_Wigner_d, [17](#)
 - phaseFactor, [17](#)
 - tripKey, [16](#)
- Ph::phChars, [46](#)
 - lam_phase, [46](#)
 - mom1, [46](#)
 - mom2, [46](#)
 - operator<, [46](#)
 - r, [46](#)
- phase
 - KFacParams, [30](#)
- phase.cc
 - Round, [77](#)
- phase.h
 - Round, [78](#)
- phaseFactor
 - Ph, [17](#)
- PolVec, [18](#)
 - getPol4, [19](#)
 - getPolz4, [19](#)
- projected
 - NPtCorr_t, [44](#)
- qm
 - KFacParams, [31](#)
- qp
 - KFacParams, [31](#)
- r
 - Ph::phChars, [46](#)
- read_xml_ini

- compute_kfactor_xml_read_write.cc, [50](#)
 - compute_kfactor_xml_read_write.h, [53](#)
- refAngles
 - LittleGrp, [14](#)
- registerAll
 - KFactorEnv, [11](#)
- Rot, [20](#)
 - eulerRotMat, [20](#)
- Round
 - phase.cc, [77](#)
 - phase.h, [78](#)
- row
 - irrep_label, [27](#)
 - IrrepLam_t, [28](#)
- smearedP
 - NPtCorr_t, [44](#)
- Sub1
 - KFacParams, [31](#)
- Sub3
 - KFacParams, [31](#)
- SubCurr
 - KFacParams, [31](#)
- 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, [83](#)
- subPhSum
 - KFacParams, [29](#)
- t_slice
 - NPtCorr_t, [45](#)
- TheKFactorFactory
 - k_factor_factory.h, [64](#)
- threeY
 - flavour, [25](#)
- toArray
 - KfUt::ToArray, [47](#)
- tripKey
 - Ph, [16](#)
- truncate
 - KfUt, [12](#)
- two_abs_lam
 - KFacParams, [30](#)
- two_lam
 - IrrepLam_t, [28](#)
- twoI
 - flavour, [25](#)
- twoIz
 - flavour, [25](#)
- twoJ
 - irrep_label, [27](#)
 - NPtCorr_t, [45](#)
- write_ei
 - compute_kfactor_xml_read_write.cc, [51](#)
 - compute_kfactor_xml_read_write.h, [54](#)
- write_irrep
 - compute_kfactor_xml_read_write.cc, [51](#)
 - compute_kfactor_xml_read_write.h, [54](#)
- write_xml_out
 - compute_kfactor_xml_read_write.cc, [51](#)
 - compute_kfactor_xml_read_write.h, [54](#)