CoherentDVCS_MC

1.0

Generated by Doxygen 1.9.1

1 Module Index

1.1 Modules

Here is a list of all modules:

2 File Index

2.1 File List

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

3 Module Documentation

3.1 Global event-by-event four-vectors

Variables

- TLorentzVector **k4Beam0**
- TLorentzVector **P4Beam0**
- TLorentzVector **Y4det**
- TLorentzVector k4Beam
- TLorentzVector **P4Beam**
- TLorentzVector **k4Scat**
- TLorentzVector **q4Virt**
- TLorentzVector **q4Prime**
- TLorentzVector **P4Scat**
- TLorentzVector **Delta4vec**
- TLorentzVector **P4Tot**

- TLorentzVector n4_e0
- TLorentzVector **n4Tilde_e0**
- TLorentzVector **X4_e0**
- TLorentzVector **Y4_e0**
- TLorentzVector **Z4_e0**
- TLorentzVector **T4_e0**
- TLorentzVector n4_e
- TLorentzVector **n4Tilde_e**
- TLorentzVector **n4_q**
- TLorentzVector **n4Tilde_q**
- TLorentzVector Y4_Det
- TLorentzVector **X4_e**
- TLorentzVector **Y4_e**
- TLorentzVector **Z4_e**
- TLorentzVector **T4_e**
- TLorentzVector X4_q
- TLorentzVector **Y4_q**
- TLorentzVector **Z4_q**
- TLorentzVector **T4_q**
- TLorentzVector **X4_pipi**
- TLorentzVector **Y4_pipi**
- TLorentzVector X4_qCM
- TLorentzVector **Y4_qCM**
- TLorentzVector **n4_qCM**
- TLorentzVector **n4Tilde_qCM**

3.1.1 Detailed Description

- $**$ Global (nominal) beam four-vectors $k_0^\mu, \quad P_0^\mu, \quad Y^\mu = [0,0,1,0] = \mathrm{VerticalUp}$

3.1.2 Variable Documentation

3.1.2.1 k4Beam `TLorentzVector k4Beam`

Event-by-event kinematic four-vectors

$$k^\mu,\, P^\mu,\, k'^\mu,\, q^\mu = (k = k')^\mu,\, P^\mu_{\pi\pi},\, P'^\mu, \Delta^\mu = (P' - P)^\mu, p^\mu_{\pi+}, p^\mu_{\pi-}$$

3.1.2.2 n4_e `TLorentzVector n4_e`

Event-by-event eP and qP light-cone four-vectors

$$n_e^\mu,\, \widetilde{n}_e^\mu,\, n_q^\mu \widetilde{n}_q^\mu$$

3.1.2.3 n4_e0 `TLorentzVector n4_e0`

Global (nominal) light-cone vectors

$$n^\mu_{e,0},\ \widetilde{n}^\mu_{e,0},\ X^\mu_{e,0},\ Y^\mu_{e,0},\ Z^\mu_{e,0},\ T^\mu_{e,0}$$

3.1.2.4 X4_q `TLorentzVector X4_q`

cartesian four-vectors

$$X^\mu_q,\ Y^\mu_q,\ Z^\mu_q,\ T^\mu_q$$

3.1.2.5 X4_qCM `TLorentzVector X4_qCM`

Event-by-event, Boosted to $q + P$ CM frame

$$X^\mu_{q,CM},\ Y^\mu_{q,CM},\ n^\mu_{q,CM},\ \widetilde{n}^\mu_{q,CM}$$

3.1.2.6 Y4_Det `TLorentzVector Y4_Det`

Event-by-event cartesian four-vectors

$$X^\mu_e,\ Y^\mu_e,\ Z^\mu_e,\ T^\mu_e$$

4 File Documentation

4.1 src/CoherentDVCS_MC.cxx File Reference

$l^A Z \to l^A Z\gamma$ lepto-production cross sections (electron or muon)

```
#include "CoherentDVCS_MC.hpp"
#include "include/Deep_event.cxx"
#include <iostream>
```

Functions

- int Init_Kin (char ∗inFile)

  Read input file to initialize Monte-Carlo event generation.
- double **GammaSmear** (TLorentzVector p4gamma, double ∗vec4out)
- int CoherentDVCS_MC (const int nEvents=1, const double tmin=-1.0)

  Monte-Carlo Driver.

4.1.1 Detailed Description

$l^A Z \rightarrow l^A Z\gamma$ lepto-production cross sections (electron or muon)

Input beam kinematics allow fixed target, head-on collisions, and crossing-angle collisions.

Created by Hyde, Charles E. on 18-June-2021.

4.2 src/CoherentDVCS_MC.hpp File Reference

File defines global variables (constants, and event-by-event variables).

```
#include <stdio.h>
#include <TCanvas.h>
#include <TH2D.h>
#include <TRandom3.h>
#include <TVector3.h>
#include <TLorentzVector.h>
#include <TH1D.h>
#include <TMath.h>
#include <TDatabasePDG.h>
```

Functions

- bool **eSmear** (false)
- bool **iSmear** (false)

Variables

- const double **TwoPi** = 2.0∗TMath::Pi()
- auto **dbPDG** = TDatabasePDG::Instance()
- TRandom3 **ran3**
- TLorentzVector **k4Beam0**
- TLorentzVector **P4Beam0**
- TLorentzVector **Y4det**
- TLorentzVector k4Beam
- TLorentzVector **P4Beam**
- TLorentzVector **k4Scat**
- TLorentzVector **q4Virt**
- TLorentzVector **q4Prime**
- TLorentzVector **P4Scat**
- TLorentzVector **Delta4vec**
- TLorentzVector **P4Tot**
- TLorentzVector n4_e0
- TLorentzVector **n4Tilde_e0**
- TLorentzVector **X4_e0**
- TLorentzVector **Y4_e0**

- TLorentzVector **Z4_e0**
- TLorentzVector **T4_e0**
- TLorentzVector n4_e
- TLorentzVector **n4Tilde_e**
- TLorentzVector **n4_q**
- TLorentzVector **n4Tilde_q**
- TLorentzVector Y4_Det
- TLorentzVector **X4_e**
- TLorentzVector **Y4_e**
- TLorentzVector **Z4_e**
- TLorentzVector **T4_e**
- TLorentzVector X4_q
- TLorentzVector **Y4_q**
- TLorentzVector **Z4_q**
- TLorentzVector **T4_q**
- TLorentzVector **X4_pipi**
- TLorentzVector **Y4_pipi**
- TLorentzVector X4_qCM
- TLorentzVector **Y4_qCM**
- TLorentzVector **n4_qCM**
- TLorentzVector **n4Tilde_qCM**
- double mLepton
- double **MIon**
- double **mPion**
- double **emitt_e** [3]
- double **emitt_i** [3]
- double **betaIP_e** [2]
- double **betaIP_i** [2]
- int **nEvents**
- double **Q2Min**
- double **Q2Max**
- double **yMin**
- double **yMax**
- double **csCMmin**
- double **csCMmax**
- double **W2Threshold**
- double **sqrtDL**
- double **deltaL**
- double **sqrtDQ**
- double **deltaQ**
- double **yInv**
- double **sMinusM2**
- double **xBj**
- double **psf**
- double **phi_e**
- double **s_e**
- double **W2**
- double **Q2**
- double **k_dot_P**
- const double **alpha** = 1.0/137.03
- const double **pi** = acos(-1.0)

- const double **umass** = 0.931494
- int **ionZ**
- int **ionA**
- const double **etaMin** = -4.0
- const double **etaEl** = -1.7
- const double **etaBe** = -1.7
- const double **etaB0** = 0.0
- const double **etaBi** = 1.4
- const double **etaMax** = 4.0
- const double **re_EMCal** = 1.2
- const double **rBarrel** = 0.6
- const double **ri_EMCal** = 2.6

4.2.1 Detailed Description

File defines global variables (constants, and event-by-event variables).

4.2.2 Variable Documentation

4.2.2.1 mLepton `double mLepton`

Invariants, definied in routine Init()

4.3 src/include/Deep_event.cxx File Reference

File generates a set of event four-vectors.

Functions

- int LeviCivita4vec (TLorentzVector vec1, TLorentzVector vec2, TLorentzVector vec3, double $*$vec4out)

  Construct a 4-vector contraction.
- double LeviCivitaScalar (TLorentzVector vec4_1, TLorentzVector vec4_2, TLorentzVector vec4_3, TLorentzVector vec4_4)
- int EventLightCone ()
- int Get_Event (int iEvt)
- double **GammaSmear** (TLorentzVector p4gamma, double $*$vec4out)

4.3.1 Detailed Description

File generates a set of event four-vectors.

File contains event-by-event generation functions for $e^A Z \to e'^A Z\gamma$.

4.3.2 Function Documentation

4.3.2.1 EventLightCone() `int EventLightCone ( )`

EventLightCone: Initialize lightcone vectors n4, n4Tilde

$$\begin{aligned}
n4 &= (\texttt{k4Beam} - \beta * \texttt{P4Beam})/norm \\
\texttt{n4Tilde} &= (\texttt{P4Beam} + \beta\text{'} * \texttt{k4Beam})/norm2; \\
n4 \cdot n4 &= 0 = n4Tilde \cdot n4Tilde \\
n4 \cdot n4Tilde &= 1 \\
k_dot_P &= k4Beam.Dot(P4Beam); \\
\texttt{sqrtDL} &= mLepton * MIon/(\texttt{k}_{\texttt{d}}\texttt{ot}_{\texttt{P}}); \\
\texttt{deltaL} &= sqrtDL * sqrtDL = \delta_L;
\end{aligned}$$

Initialize lightcone vectors n4_e, n4Tilde_e, X4_e, Y4_e.

$$\begin{aligned}
n_e^\mu &= \frac{\left[k^\mu \left(1+\sqrt{1-\delta_l}\right) - \frac{m_l^2}{k\cdot P} P^\mu\right]}{\left[2\sqrt{(k\cdot P)(1-\delta_l)}\right]} \\
k^2 &= m_l^2, \qquad \delta_l = \left[\frac{m_l M_{\rm Ion}}{k\cdot P}\right]^2 \\
\widetilde{n}_e^\mu &= \left[P^\mu - \frac{M_{\rm Ion}^2}{(k\cdot P)\left(1+\sqrt{1-\delta_l}\right)} k^\mu\right] \Bigg/ \frac{1}{\sqrt{(k\cdot P)(1-\delta_l)}} \\
n_e \cdot n_e &= 0 = \widetilde{n}_e \cdot \widetilde{n}_e \\
n_e \cdot \widetilde{n}_e &= 1 \qquad (1)
\end{aligned}$$

Transverse vectors

$$\begin{aligned}
\texttt{Y4Det} &= [0,0,1,0] = Y4^\mu_{\textsf{Det}} = \text{“up in Detector frame"} \\
[X4_{e0}]_\sigma &= \epsilon_{\mu\nu\rho\sigma} n4^\mu_e n4Tilde^\nu_e Y4_D et^\rho \\
X4^\mu_e &= X4^\mu_{e0}/\sqrt{-X4_{e0}\cdot X4_{e0}} \\
[Y4_e]_\sigma &= \epsilon_{\mu\nu\rho\sigma} n4^\mu_e X4^\nu_e n4Tilde^\rho_e \qquad (2) \\
& \qquad (3)
\end{aligned}$$

If beam emmitance values in input file are positive, incident beam 4-vectors k^μ, P^μ are generated with gaussian longitudinal and transverse emmittance relative to nominal input values k_0^μ, P_0^μ.

The transverse 4-vectors X_e^μ, Y_e^μ are defined assuming neither incident beam can ever be in the vertical direction.

4.3.2.2 Get_Event() `int Get_Event ( int iEvt )`

Get_Event() generates e Z->e Z gammai events uniformly in phase space

$$\{Q^2, y = q \cdot P/(k \cdot P), \phi_e, \cos(\theta_{\gamma\gamma}^{CM}), \phi_{\gamma\gamma}^{CM}$$

Only basis 4-vectors e.g. $n_q, \widetilde{n}_q, X_q, Y_q$ are boosted, all other variables are invariants.

After smearing the incident beam momenta, re-define the lepton-Ion light-cone vectors n4_e, n4Tilde_e, X4_e, Y4_e

Scattered lepton four-vector k4Scat = k'.

$$\begin{bmatrix} Q^2 + 2m_l^2 \\ 2(k \cdot P)(1-y) \end{bmatrix} = 2 \begin{bmatrix} k \cdot \widetilde{n}_e, & k \cdot n_e \\ P \cdot \widetilde{n}_e, & P \cdot n_e \end{bmatrix} \begin{bmatrix} k' \cdot n_e \\ k' \cdot \widetilde{n}_e \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} k' \cdot n_e \\ k' \cdot \widetilde{n}_e \end{bmatrix} = \frac{1}{(k \cdot \widetilde{n}_e)(P \cdot n_e) - (k \cdot n_e)(P \cdot \widetilde{n}_e)} \begin{bmatrix} P \cdot n_e, & -k \cdot n_e \\ -P \cdot \widetilde{n}_e, & k \cdot \widetilde{n}_e \end{bmatrix} \begin{bmatrix} Q^2/2 + m_l^2 \\ (k \cdot P)(1-y) \end{bmatrix} \tag{5}$$

$$\mathbf{k}_\perp'^2 = 2(k' \cdot n_e)(k' \cdot \widetilde{n}_e) - m_l^2 \tag{6}$$

Define the lightcone vectors of q + P system

$$n_q^\mu = \frac{1}{\sqrt{2(1+\delta_Q)}} \left[q^\mu \frac{M_{Ion}}{q \cdot P} + P^\mu \frac{\delta_Q}{M_{Ion}\left(1 + \sqrt{1+\delta_Q}\right)} \right], \qquad \delta_Q = \frac{Q^2 M_{\text{Ion}}^2}{(q \cdot P)^2} \tag{7}$$

$$\widetilde{n}_q^\mu = \frac{1}{\sqrt{2(1+\delta_Q)}} \left[P^\mu \frac{\left(1 + \sqrt{1+\delta_Q}\right)}{M_{\text{Ion}}} - q^\mu \frac{M_{\text{Ion}}}{q \cdot P} \right] \tag{8}$$

Recoil proton kinematics:

- After boosting to the $q + P$ CM frame, X_q^μ, Y_q^μ are pure space-like (vanishing time components) and the space parts of n_q^μ, $\widetilde{n}_q^\mu$ are anti-colinear.
- In this frame, the incident proton is collinear with the unit three vector
- $\mathbf{v} = [\widetilde{n}_q.Px(), \widetilde{n}_q.Py(), \widetilde{n}_q.Pz()]/\,\widetilde{n}_q.E()$
  - The recoil nucleon and the $\pi\pi$ system are back-to-back in $q + P$ CM frame.
- Generate uniform distributions in
  - cosine of polar-angle: $\cos\left(\theta_{\pi\pi}^{CM}\right)$; and
  - azimuthal-angle $\phi_{\pi\pi}^{CM}$
- $$\begin{bmatrix} E' \\ |\mathbf{P}'| \end{bmatrix}^{CM} = \begin{bmatrix} (W^2 + M_{\text{Ion}}^2 - M_{\pi\pi}^2)/(2\sqrt{W^2}) \\ \sqrt{[E'^{\text{CM}}]^2 - M_{\text{Ion}}^2} \end{bmatrix}$$

  Construct CM unit 3-vector in incident ion direction

  $$\mathbf{v} = \left[\frac{\widetilde{n}_q.Px()}{\widetilde{n}_q.E()}, \frac{\widetilde{n}_q.Py()}{\widetilde{n}_q.E()}, \frac{\widetilde{n}_q.Pz()}{\widetilde{n}_q.E()} \right]^{CM}$$

  Define a longitudinal four-vector of the recoil proton in the CM frame:

  $$\left[P_L'^\mu\right]^{CM} = \left[E', -|\mathbf{P}'| \cos\left(\theta_{\pi\pi}^{\text{CM}}\right) \mathbf{v}\right]^{\text{CM}}$$

  In any frame, the full recoil proton momentum four-vector is defined by:

  $$P'^\mu = \left(P_L'^{CM} \cdot n_q^{CM}\right) \widetilde{n}_q^\mu + \left(P_L'^{CM} \cdot \widetilde{n}_q^{CM}\right) n_q^\mu - |\mathbf{P}'|^{\text{CM}} \sin\theta_{\pi\pi}^{CM} \left[\cos\phi_{\pi\pi}^{CM} X_q^\mu + \sin\phi_{\pi\pi}^{CM} Y_q^\mu\right]$$

  The momentum four-vector of the $\pi\pi$ system is simply

  $$P_{\pi\pi}^\mu = q^\mu + P^\mu - P'^\mu$$

4.3.2.3 LeviCivita4vec()

```
int LeviCivita4vec (
    TLorentzVector vec1,
    TLorentzVector vec2,
    TLorentzVector vec3,
    double * vec4out )
```

Construct a 4-vector contraction.

$$\begin{gathered}\texttt{vec4out}_{[0,1,2,3]} = \texttt{vec4out}_{\mu} = \epsilon_{\mu\nu\rho\sigma}\texttt{vec1}^{\nu}\texttt{vec2}^{\rho}\texttt{vec3}^{\sigma} \\ \epsilon_{0123} = 1\end{gathered} \tag{9}$$

4.3.2.4 LeviCivitaScalar()

```
double LeviCivitaScalar (
    TLorentzVector vec4_1,
    TLorentzVector vec4_2,
    TLorentzVector vec4_3,
    TLorentzVector vec4_4 )
```

Construct a the scaler anti-symmetric contraction of four space-time vectors

$$*\texttt{Scalar} = \epsilon_{\mu\nu\rho\sigma}\texttt{vec4}_1{}^{\mu}\texttt{vec4}_2{}^{\nu}\texttt{vec4}_3{}^{\rho}\texttt{vec4}_4{}^{\sigma} \tag{10}$$

$$\epsilon_{0123} = 1. \qquad \text{In TLorentzVector Notation} \quad \epsilon_{xyzt} = -\epsilon_{0123} = -1 \tag{11}$$

$$\tag{12}$$

Note from TLorentzVector documentation The components of TLorentzVector can also accessed by index:

```
xx = v(0);      or     xx = v[0];
yy = v(1);             yy = v[1];
zz = v(2);             zz = v[2];
tt = v(3);             tt = v[3];
```

4.4 src/include/ScatAmp.cxx File Reference

File containing functions to calculate scattering amplitude for $ep \rightarrow e'p'\pi\pi$.

Functions

- double LeptonSymmTensor (int lambda_q1, int lambda_q2)
- double LeptonAntiTensor (int lambda_q1, int lambda_q2)

4.4.1 Detailed Description

File containing functions to calculate scattering amplitude for $ep \rightarrow e'p'\pi\pi$.

4.4.2 Function Documentation

4.4.2.1 LeptonAntiTensor()

```
double LeptonAntiTensor (
        int lambda_q1,
        int lambda_q2 )
```

Calculate the anti-symmetric lepton tensor Virtual photon helicities: lambda_q1, lambda_q2 $= \lambda_q(i) \in -1, 0, 1$

4.4.2.2 LeptonSymmTensor()

```
double LeptonSymmTensor (
        int lambda_q1,
        int lambda_q2 )
```

Calculate the symmetric lepton tensor Virtual photon helicities: lambda_q1, lambda_q2 $= \lambda_q(i) \in -1, 0, 1$

4.5 src/include/ScatAmp.hxx File Reference

Header file declaring functions to calculate scattering amplitude for $ep \rightarrow e'p'\pi\pi$.

Functions

- double **Diffract** ()
- double **piN** ()
- double **LeptonSymmTensor** ()
- double **LeptonAntiTensor** ()

4.5.1 Detailed Description

Header file declaring functions to calculate scattering amplitude for $ep \rightarrow e'p'\pi\pi$.