LPAIR++

0.1

Generated by Doxygen 1.8.3.1

Fri Apr 26 2013 14:17:37

Contents

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

2 Data Structure Documentation

2.1 Cuts Class Reference

List of kinematic cuts to apply on the central and outgoing phase space.

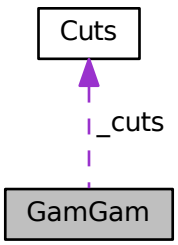
Data Fields

- double emax
  Maximal energy of the central two-photons system.
- double emin
  Minimal energy of the central two-photons system.
- int mode
  Sets of cuts to apply on the final phase space.
- double ptmax
  Maximal transverse momentum of the single outgoing leptons.
- double ptmin
  Minimal transverse momentum of the single outgoing leptons.
- double thetamax
  Maximal polar (θ) angle of the outgoing leptons.
- double thetamin
  Minimal polar (θ) angle of the outgoing leptons.

2.2 GamGam Class Reference

Computes the matrix element for a $\gamma\gamma \to \ell^+\ell^-$ process.

Collaboration diagram for GamGam:



Public Member Functions

- GamGam (int, double, double, int, double x_[])

  Class constructor.
- void ComputeSqS ()

  Computes $\sqrt{s}$ for the system.
- double ComputeXsec (int nm_=1)

  Computes the process' cross section.
- Particle **GetParticle** (int)
- bool IsKinematicsDefined ()

  Is the system's kinematics well defined?
- bool Orient ()

  Energies/momenta computation for the various particles, in the CM system.
- double PeriPP (int, int)

  Computes the matrix element squared for the requested process.
- bool Pickin ()
- void SetCuts (Cuts)

  Sets the list of kinematic cuts to apply on the outgoing particles' final state.
- bool SetIncomingKinematics (int, double[], int)

  Sets the momentum and PDG id for the incoming particles.
- bool SetOutgoingParticles (int, int)

  Sets the PDG id for the outgoing particles.
- void SetWRange (double, double)

  Sets the energy range available for the phase space integration.

Private Attributes

- double **_a1**
- double **_a2**
- double **_a5**
- double **_a6**
- double **_acc3**
- double **_acc4**
- double **_al3**
- double **_al4**
- double **_bb**
- double **_be4**
- double **_be5**
- double _betgam

  $\beta\gamma$ factor of the centre of mass system, used in the computation of the inverse boost for the outgoing leptons
- double **_cotth1**
- double **_cotth2**
- double _cp3

  $\cos\phi_3$ of the first outgoing proton-like particle
- double _cp5

  $\cos\phi_5$ of the second outgoing proton-like particle
- double _cp6

  $\cos\phi_6$ of the first outgoing lepton
- double _cp7

  $\cos\phi_7$ of the second outgoing lepton
- double _ct3

  $\cos\theta_3$ of the first outgoing proton-like particle

- double _ct4

  $\cos\theta_4$ *of the two-photons centre of mass system*
- double _ct5

  $\cos\theta_5$ *of the second outgoing proton-like particle*
- double _ct6

  $\cos\theta_6$ *of the first outgoing lepton*
- double _ct7

  $\cos\theta_7$ *of the second outgoing lepton*
- double _ctcm6

  $\cos\theta_6^{\rm CM}$*, production angle of the first outgoing lepton, computed in the centre of mass system.*
- Cuts _cuts

  Set of cuts to apply on the final phase space.
- double _d1

  $\delta_1 = m_3^2 - m_1^2$ *as defined in Vermaseren's paper*
- double **_d1dq**
- double **_d1dq2**
- double _d2

  $\delta_4 = m_5^2 - m_2^2$ *as defined in Vermaseren's paper*
- double **_d3**
- double _d4

  $\delta_5 = m_4^2 - t_1$ *as defined in Vermaseren's paper*
- double _d5

  $\delta_2 = m_1^2 - m_2^2$ *as defined in Vermaseren's paper*
- double _d6

  $\delta_6 = m_4^2 - m_5^2$ *as defined in Vermaseren's paper*
- double **_d7**
- double _d8

  $\delta_3 = t_1 - m_2^2$ *as defined in Vermaseren's paper*
- double **_dd1**
- double **_dd2**
- double **_dd3**
- double **_dd4**
- double **_dd5**
- double **_de3**
- double **_de5**
- double **_delta**
- double **_dj**
- double _e6lab

  $E_6^{\rm lab}$*, energy of the first outgoing lepton, computed in the lab frame*
- double _e7lab

  $E_7^{\rm lab}$*, energy of the second outgoing lepton, computed in the lab frame*
- double _ec4

  E_4*, energy of the two-photon central system*
- double **_ecut**
- double _el6

  E_6*, energy of the first outgoing lepton*
- double _el7

  E_7*, energy of the second outgoing lepton*
- double _ep1

  E_1*, energy of the first proton-like incoming particle*
- double _ep2

  E_2*, energy of the second proton-like incoming particle*

- double _ep3

  E_3, *energy of the first proton-like outgoing particle*
- double _ep5

  E_5, *energy of the second proton-like outgoing particle*
- double **_epsi**
- double _etot

  Total energy provided by the two incoming proton-like particles.
- double **_g4**
- double **_g5**
- double **_g6**
- double _gamma

  γ *factor of the centre of mass system, used in the computation of the inverse boost for the outgoing leptons*
- double **_gram**
- double _mc4

  m_4, *mass of the two-photon central system*
- double _ml6

  m_6, *mass of the first outgoing lepton*
- double _ml7

  m_7, *mass of the second outgoing lepton*
- double _mp1

  m_1, *mass of the first proton-like incoming particle*
- double _mp2

  m_2, *mass of the second proton-like incoming particle*
- double _mp3

  m_3, *mass of the first proton-like outgoing particle*
- double _mp5

  m_5, *mass of the second proton-like outgoing particle*
- int _ndim

  Number of dimensions on which the integration has to be performed.
- int **_nOpt**
- double **_p**
- double **_p12**
- double **_p13**
- double **_p14**
- double **_p15**
- double **_p1k2**
- double **_p23**
- double **_p24**
- double **_p25**
- double **_p2k1**
- double **_p34**
- double **_p35**
- double _p3_c4 [3]

  $\mathbf{p}_4$, *3-momentum of the two-photon central system*
- double _p3_l6 [3]

  $\mathbf{p}_6$, *3-momentum of the first outgoing lepton*
- double _p3_l7 [3]

  $\mathbf{p}_7$, *3-momentum of the second outgoing lepton*
- double _p3_p1 [3]

  $\mathbf{p}_1$, *3-momentum of the first proton-like incoming particle*
- double _p3_p2 [3]

  $\mathbf{p}_2$, *3-momentum of the second incoming particle*

- double _p3_p3 [3]

  $\mathbf{p}_3$, *3-momentum of the first proton-like outgoing particle*
- double _p3_p5 [3]

  $\mathbf{p}_5$, *3-momentum of the second proton-like outgoing particle*
- double **_p45**
- double **_p_p3**
- double **_p_p4**
- double **_p_p5**
- double _pc4

  $|\mathbf{p}_4|$, *3-momentum norm of the two-photon central system*
- int **_pdg1**
- int **_pdg2**
- int **_pdg3**
- int **_pdg5**
- int **_pdg6**
- int **_pdg7**
- double _pl6

  $|\mathbf{p}_6|$, *3-momentum norm of the first outgoing lepton*
- double _pl7

  $|\mathbf{p}_7|$, *3-momentum norm of the second outgoing lepton*
- double _pp1

  $|\mathbf{p}_1|$, *3-momentum norm of the first proton-like incoming particle*
- double _pp2

  $|\mathbf{p}_2|$, *3-momentum norm of the second proton-like incoming particle*
- double _pp3

  $|\mathbf{p}_3|$, *3-momentum norm of the first proton-like outgoing particle*
- double _pp5

  $|\mathbf{p}_5|$, *3-momentum norm of the second proton-like outgoing particle*
- double _pt_l6

  $p_{T,6}$, *transverse momentum of the first outgoing lepton*
- double _pt_l7

  $p_{T,7}$, *transverse momentum of the second outgoing lepton*
- double **_ptcut**
- double _ptot

  Total momentum provided by the two incoming proton-like particles (along the z-axis)
- double **_q1dq**
- double **_q1dq2**
- double _q2max

  Maximal Q^2 exchange.
- double _q2min

  Minimal Q^2 exchange.
- double **_qp2max**
- double **_qp2min**
- double **_qve** [4]
- double _s

  s, *squared centre of mass energy of the incoming particles' system*
- double **_s1**
- double **_s2**
- double **_sa1**
- double **_sa2**
- double **_sl1**
- double _sp3

$\sin\phi_3$ *of the first outgoing proton-like particle*

- double _sp5

  $\sin\phi_5$ *of the second outgoing proton-like particle*
- double _sp6

  $\sin\phi_6$ *of the first outgoing lepton*
- double _sp7

  $\sin\phi_7$ *of the second outgoing lepton*
- double _sqs

  $\sqrt{s}$, *centre of mass energy of the incoming particles' system*
- double _st3

  $\sin\theta_3$ *of the first outgoing proton-like particle*
- double _st4

  $\sin\theta_4$ *of the two-photons centre of mass system*
- double _st5

  $\sin\theta_5$ *of the second outgoing proton-like particle*
- double _st6

  $\sin\theta_6$ *of the first outgoing lepton*
- double _st7

  $\sin\theta_7$ *of the second outgoing lepton*
- double _stcm6

  $\sin\theta_6^{\mathrm{CM}}$, *production angle of the first outgoing lepton, computed in the centre of mass system.*
- double **_t1**
- double **_t2**
- double **_tau**
- double _w1

  m_1^2, *squared mass of the first proton-like incoming particle*
- double **_w12**
- double _w2

  m_2^2, *squared mass of the second proton-like incoming particle*
- double _w3

  m_3^2, *squared mass of the first proton-like outgoing particle*
- double **_w31**
- double _w4

  m_4^2, *squared mass of the two-photon central system*
- double _w5

  m_5^2, *squared mass of the second proton-like outgoing particle*
- double **_w52**
- double _w6

  m_6^2, *squared mass of the first outgoing lepton*
- double _w7

  m_7^2, *squared mass of the second outgoing lepton*
- double **_wmax**
- double **_wmin**
- double * _x

  Array of _ndim components representing the point on which the weight in the cross-section is computed.
- bool setin

  Is the incoming particles' kinematic set ?
- bool setkin

  Is the full event's kinematic set ?
- bool setll

  Is the outgoing leptons' state set ?

- bool setout

  Is the outgoing particles' kinematic set ?
- bool setp1

  Is the first incoming proton-like particle's kinematic set ?
- bool setp2

  Is the second incoming proton-like particle's kinematic set ?
- bool setp3

  Is the first outgoing proton-like particle's kinematic set ?
- bool setp5

  Is the second outgoing proton-like particle's kinematic set ?

2.2.1 Constructor & Destructor Documentation

2.2.1.1 GamGam::GamGam (int *ndim_*, double *q2min_*, double *q2max_*, int *nOpt_*, double *x_[]*)

Sets the mandatory parameters used in the methods computing the kinematics and the cross-section of this phase space point.

Parameters

*ndim_*	The number of dimensions of the point in the phase space
*q2min_*	The minimal value of Q^2
*q2max_*	The maximal value of Q^2
x_[]	The ndim_-dimensional point in the phase space on which the kinematics and the cross-section are computed

2.2.2 Member Function Documentation

2.2.2.1 void GamGam::ComputeSqS ( )

Computes the centre of mass energy for the system, according to the incoming particles' kinematics

Here is the caller graph for this function:

2.2.2.2 double GamGam::ComputeXsec (int *nm_* = 1)

Computes the cross-section for the $\gamma\gamma \to \ell^+\ell^-$ process with the given kinematics

Returns

$\sigma_{\gamma\gamma\to\ell^+\ell^-}$, the total cross section for the given event, provided the phase space region on which it is integrated.

2.2.2.3 bool GamGam::IsKinematicsDefined ( ) [inline]

Is the system's kinematics well defined? Mandatory to perform the point's cross-section computation.

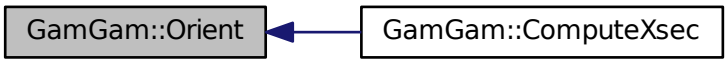
Returns

A boolean stating if the input kinematics and the final states are well defined

2.2.2.4 bool GamGam::Orient ( )

Calculates energies and momenta of the 1st, 2nd, 3rd, 4th and 5th particle in the overall centre of mass frame.

Here is the caller graph for this function:



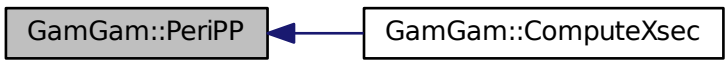
2.2.2.5 double GamGam::PeriPP (int *nup_*, int *ndown_*)

Contains the expression of the matrix element squared for the process under considerations. It returns the value of the product of the form factor or structure functions and the matrix element squared.

Returns

The full matrix element for the two-photon production of a pair of spin $-\frac{1}{2}-$point particles

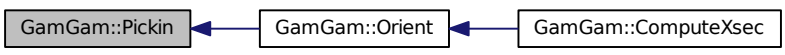
Here is the caller graph for this function:



2.2.2.6 bool GamGam::Pickin ( )

Describes the kinematics of the process $p_1 + p_2 \to p_3 + p_4 + p_5$ in terms of Lorentz-invariant variables. Fills several variables to be used in PeriPP (essential for the evaluation of the matrix element).

Here is the caller graph for this function:



2.2.2.7 void GamGam::SetCuts (Cuts *cuts_*)

Parameters

*cuts_*	The Cuts object containing the kinematic parameters

2.2.2.8 bool GamGam::SetIncomingKinematics (int , double *[]*, int)

Parameters

*part_*	Role of the particle in the process
*momentum_*	3-momentum of the particle
*pdgId_*	Particle ID according to the PDG convention

2.2.2.9 bool GamGam::SetOutgoingParticles (int *part_*, int *pdgId_*)

Parameters

*part_*	Role of the particle in the process
*pdgId_*	Particle ID according to the PDG convention

2.2.2.10 void GamGam::SetWRange (double *wmin_*, double *wmax_*)

Parameters

*wmin_*	The minimal s on which the cross section is integrated
*wmin_*	The maximal s on which the cross section is integrated. If negative, the maximal energy available to the system (hence, $s = (\sqrt{s})^2$) is provided.

2.2.3 Field Documentation

2.2.3.1 double GamGam::_d1 [private]

[2] for the full definition of this quantity

2.2.3.2 double GamGam::_d2 [private]

[2] for the full definition of this quantity

2.2.3.3 double GamGam::_d4 [private]

[2] for the full definition of this quantity

2.2.3.4 double GamGam::_d5 [private]

[2] for the full definition of this quantity

2.2.3.5 double GamGam::_d6 [private]

[2] for the full definition of this quantity

2.2.3.6 double GamGam::_d8 [private]

[2] for the full definition of this quantity

2.3 Gnuplot Class Reference

Public Member Functions

- **Gnuplot** (std::string outFile_="")

- int **DrawHistogram** ()
- int **Fill** (double)
- void operator<< (const std::string &command)

  Feeds a command line to the Gnuplot interpreter.
- void **SetHistogram** (int, double, double)
- void SetLogy (bool logy_=true)

  Toggles the logarithmic scale for the y-axis.
- void SetOutputFile (std::string)
- void SetTitle (std::string)

  Sets the title for the graph.
- void SetXAxisTitle (std::string)

  Sets the caption for the x-axis.
- void SetYAxisTitle (std::string)

  Sets the caption for the y-axis.

Protected Attributes

- FILE * _pipe

  The pipe used to feed the Gnuplot interpreter.

Private Attributes

- double * **_histBounds**
- double **_histHigh**
- double **_histLow**
- int **_histNum**
- int **_histOverflow**
- int **_histUnderflow**
- int * **_histValues**
- bool **_isHist**

2.3.1 Member Function Documentation

2.3.1.1 void Gnuplot::operator<< (const std::string & *command*)

Parameters

&command	The Gnuplot-formatted command line to feed

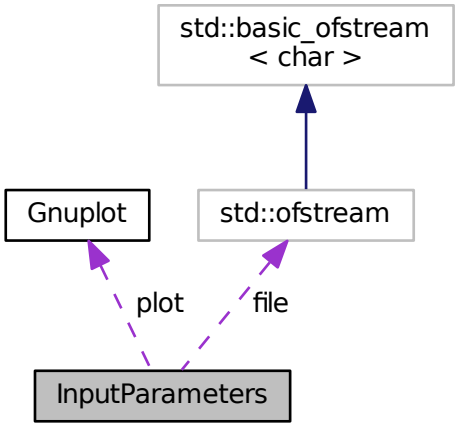
2.3.1.2 void Gnuplot::SetOutputFile (std::string *outFile_*)

Sets the file on which the graph has to be produced

2.4 InputParameters Class Reference

List of input parameters used to start and run the simulation job.

Collaboration diagram for InputParameters:



Data Fields

- bool **debug**
- std::ofstream * file

  The file in which to store the events generation's output.
- bool generation

  Are we generating events ? (true) or are we only computing the cross-section ? (false)
- double in1p

  First incoming particle's momentum (in GeV/c)
- double in2p

  Second incoming particle's momentum (in GeV/c)
- int itvg

  Number of Vegas integrations.
- double maxpt

  Maximal transverse momentum of the outgoing leptons.
- int mcut

  Set of cuts to apply on the outgoing leptons.
- double minpt

  Minimal transverse momentum of the outgoing leptons.
- int **ncvg**
- int p1mod

  First particle's mode.
- int p2mod

  Second particle's mode.
- int pair

  PDG id of the outgoing leptons.
- Gnuplot * **plot**

2.4.1 Field Documentation

2.4.1.1 int InputParameters::mcut

Set of cuts to apply on the outgoing leptons in order to restrain the available kinematic phase space :

- 0 - No cuts at all (for the total cross section)
- 1 - Vermaserens' hypothetical detector cuts : for both leptons,
  - $\frac{|p_z|}{|\mathbf{p}|} \leq 0.75$ and $p_T \geq 1$ GeV, or
  - $0.75 < \frac{|p_z|}{|\mathbf{p}|} \leq 0.95$ and $p_z > 1$ GeV,
- 2 - Cuts according to the provided parameters

2.4.1.2 int InputParameters::p1mod

The first incoming particle type and kind of interaction :

- 1 - electron,
- 2 - proton elastic,
- 3 - proton inelastic without parton treatment,
- 4 - proton inelastic in parton model

Note

Was named PMOD in ILPAIR

2.4.1.3 int InputParameters::p2mod

Note

Was named EMOD in ILPAIR

2.4.1.4 int InputParameters::pair

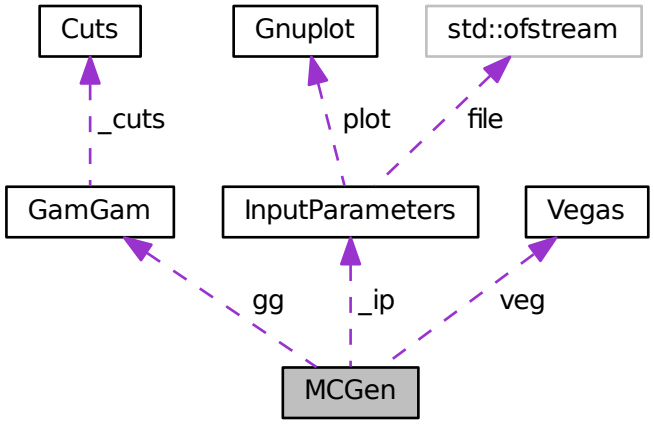
The particle code of produced leptons :

- 11 - e^+e^-
- 13 - $\mu^+\mu^-$
- 15 - $\tau^+\tau^-$

2.5 MCGen Class Reference

Core Monte-Carlo generator.

Collaboration diagram for MCGen:



Public Member Functions

- MCGen (InputParameters)
  Class constructor.
- void **AnalyzePhaseSpace** (std::string)
- InputParameters GetInputParameters ()
  Returns the set of parameters used to setup the phase space to integrate.
- void **LaunchGen** (int)

Private Attributes

- int **_inp1pdg**
- int **_inp2pdg**
- InputParameters _ip
  Set of parameters to setup the phase space to integrate.
- int _ndim
  Number of dimensions on which to perform the integration.
- int **_outp1pdg**
- int **_outp2pdg**
- GamGam * gg
  The GamGam object computing the kinematics and the cross-section for the given point in the phase space.
- Vegas * veg
  The Vegas integrator which will integrate the function.

2.5.1 Constructor & Destructor Documentation

2.5.1.1 MCGen::MCGen (InputParameters *ip_*)

Sets the number of dimensions on which to perform the integration, according to the set of input parameters given as an argument and propagated to the whole object

Parameters

*ip_*	List of input parameters defining the phase space on which to perform the integration

2.5.2 Member Function Documentation

2.5.2.1 InputParameters MCGen::GetInputParameters ( ) [inline]

Returns

The InputParameter object embedded in this class

2.6 Particle Class Reference

Kinematics of one particle.

Public Member Functions

- std::string GetLHEline ()
- void SetP (double, double, double)

  Sets the momentum.

Data Fields

- double e

  Energy.
- double m

  Mass.
- int pdgId

  Particle Data Group integer identifier.
- double pt

  Transverse momentum.
- double px

  Momentum along the x-axis.
- double py

  Momentum along the y-axis.
- double pz

  Momentum along the z-axis.
- int role

  Role in the considered process.

2.6.1 Detailed Description

Kinematic information for one particle

2.6.2 Member Function Documentation

2.6.2.1 std::string Particle::GetLHEline ( )

Returns a string containing all the particle's kinematics as expressed in the Les Houches format

Returns

The LHE line

2.6.2.2 void Particle::SetP (double *px_*, double *py_*, double *pz_*)

Parameters

*px_*	Momentum along the x-axis
*py_*	Momentum along the y-axis
*pz_*	Momentum along the z-axis

2.7 Vegas Class Reference

Vegas Monte-Carlo integrator instance.

Public Member Functions

- Vegas (int, double f_(double *, size_t, void *), InputParameters *inParam_)
- ~Vegas ()
  Class destructor.
- int LaunchGeneration (int)
  Launches the generation of events.
- int LaunchIntegration ()
  Launches the integration of the provided function.

Private Attributes

- gsl_monte_function * _F
  The wrapped-up function to integrate, along with the input parameters.
- size_t _ncalls
  Fixed number of function calls to use.
- size_t _ndim
  The number of dimensions on which to integrate the function.
- size_t _nIter
  Number of points to generate in order to integrate the function.
- gsl_rng * _r
  GSL's random number generator.
- gsl_monte_vegas_state * _s
  GSL's Vegas integration state structure.
- double * _xl
  Lower bounds for the points to generate.
- double * _xu
  Upper bounds for the points to generate.

2.7.1 Constructor & Destructor Documentation

2.7.1.1 Vegas::Vegas (int *dim_*, double *f_double *, size_t, void *, InputParameters * *inParam_*)

Constructs the class by booking the memory and structures for the GSL Vegas integrator. This code from the GNU scientific library is based on the Vegas Monte Carlo integration algorithm developed by P. Lepage. [1]

Parameters

*dim_*	The number of dimensions on which the function will be integrated
*f_*	The function one is required to integrate
*inParam_*	A list of parameters to define the phase space on which this integration is performed (embedded in an InputParameters object)

2.7.2 Member Function Documentation

2.7.2.1 int Vegas::LaunchGeneration (int *nEvts_*)

Launches the Vegas generation of events according to the provided input parameters.

Parameters

*nEvts_*	The number of events to generate

2.7.2.2 int Vegas::LaunchIntegration ( )

Launches the Vegas integration of the provided function with the provided input parameters.

References

[1] G Peter Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192 – 203, 1978. 16

[2] J.A.M. Vermaseren. Two-photon processes at very high energies. *Nuclear Physics B*, 229(2):347 – 371, 1983. 10

Index