LPAIR++ 0.2

Generated by Doxygen 1.8.6

Wed Apr 16 2014 22:06:02

ii CONTENTS

Contents

1	Princ	ciples	1
2	Todo	o List	1
3	Depi	recated List	2
4	Hiera	archical Index	2
	4.1	Class Hierarchy	2
5	Data	a Structure Index	2
	5.1	Data Structures	2
6	Data	a Structure Documentation	3
	6.1	Event Class Reference	3
		6.1.1 Detailed Description	5
		6.1.2 Member Function Documentation	5
	6.2	GamGamLL Class Reference	9
		6.2.1 Detailed Description	16
		6.2.2 Constructor & Destructor Documentation	17
		6.2.3 Member Function Documentation	18
	6.3	GamGamWW Class Reference	21
		6.3.1 Member Function Documentation	23
	6.4	GamPomVMLL Class Reference	24
		6.4.1 Member Function Documentation	27
		6.4.2 Field Documentation	30
	6.5	Hadroniser Class Reference	31
		6.5.1 Detailed Description	32
		6.5.2 Member Function Documentation	32
	6.6	HEPEUP Class Reference	33
	6.7	HEPRUP Class Reference	34
		6.7.1 Detailed Description	35
	6.8	Herwig6Hadroniser Class Reference	35
		6.8.1 Member Function Documentation	36
	6.9	Jetset7Hadroniser Class Reference	36
		6.9.1 Member Function Documentation	38
	6.10	Kinematics Class Reference	38
		6.10.1 Field Documentation	39
	6.11	MCGen Class Reference	40
	U.11	6.11.1 Detailed Description	41
		6.11.2 Constructor & Destructor Documentation	41
		VIII Constructor & Destructor Documentation	

1 Principles 1

	6.11.3 Member Function Documentation	41
6.12	Parameters Class Reference	42
	6.12.1 Detailed Description	44
	6.12.2 Member Function Documentation	44
	6.12.3 Field Documentation	45
6.13	Particle Class Reference	46
	6.13.1 Detailed Description	48
	6.13.2 Member Function Documentation	48
	6.13.3 Field Documentation	53
6.14	Process Class Reference	53
	6.14.1 Detailed Description	55
	6.14.2 Member Function Documentation	55
6.15	Pythia6Hadroniser Class Reference	57
	6.15.1 Detailed Description	58
	6.15.2 Member Function Documentation	58
6.16	Pythia8Hadroniser Class Reference	59
	6.16.1 Member Function Documentation	60
6.17	Vegas Class Reference	60
	6.17.1 Detailed Description	62
	6.17.2 Constructor & Destructor Documentation	62
	6.17.3 Member Function Documentation	62
Bibliogr	aphy	64
Index		65

1 Principles



This Monte Carlo generator, based on the LPAIR code developed in the early 1990s by J. Vermaseren *et al*[4], allows to compute the cross-section and to generate events for the $\gamma\gamma \to \ell^+\ell^-$ process in high energy physics.

The main operation is the integration of the matrix element (given as a subset of a Process object) performed by *Vegas*, an importance sampling algorithm written in 1972 by G. P. Lepage[2].

2 Todo List

Global GamGamLL::ComputeWeight ()

Find out what this *nm*_ parameter does...

Global GamGamLL::GamGamLL (int nOpt_=0)

Figure out how this nOpt_ parameter is affecting the final cross-section computation and events generation

3 Deprecated List

Global MCGen::LaunchGeneration ()

This method is to be suppressed since the events generation can now be launched one event at a time using the *GenerateOneEvent* method

4 Hierarchical Index

4.1 Class Hierarchy

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

Event	3
Hadroniser	31
Herwig6Hadroniser	35
Jetset7Hadroniser	36
Pythia6Hadroniser	57
Pythia8Hadroniser	59
HEPEUP	33
HEPRUP	34
Kinematics	38
MCGen	40
Parameters	42
Particle	46
Process	53
GamGamLL	9
GamGamWW	21
GamPomVMLL	24
Vegas	60

5 Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

Event Kinematic information on the particles in the event	3
GamGamLL Computes the matrix element for a CE $\gamma\gamma \to \ell^+\ell^-$ process	9

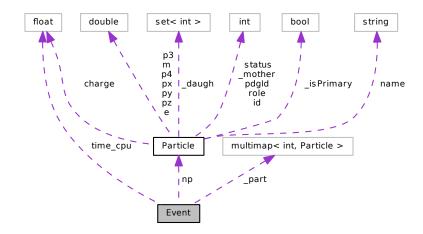
GamGamWW Computes the matrix element for a CE $\gamma\gamma\to W^+W^-$ process	21
$\label{eq:computes} \begin{tabular}{ll} \textbf{GamPomVMLL} \\ \textbf{Computes the matrix element for a CE } $\gamma \mathbb{P} \to J/\psi, \Upsilon \to \ell^+\ell^-$ process \end{tabular}$	24
Hadroniser	31
HEPEUP User-process event information	33
HEPRUP Generic user-process interface for events generator	34
Herwig6Hadroniser Herwig6 hadronisation algorithm	35
Jetset7Hadroniser Jetset7 hadronisation algorithm	36
Kinematics List of kinematic cuts to apply on the central and outgoing phase space	38
MCGen Core of the Monte-Carlo generator	40
Parameters List of parameters used to start and run the simulation job	42
Particle Kinematics of one particle	46
Process	5 3
Pythia6Hadroniser Pythia6 hadronisation algorithm	57
Pythia8Hadroniser	5 9
Vegas Vegas Monte-Carlo integrator instance	6 0

6 Data Structure Documentation

6.1 Event Class Reference

Kinematic information on the particles in the event.

Collaboration diagram for Event:



Public Member Functions

- int AddParticle (Particle *part_, bool replace_=false)
 - Add a particle to the event.
- int AddParticle (int role_, bool replace_=false)
- void clear ()
- void Dump (bool stable_=false)
- Particle * GetById (int id_)

Gets one particle by its unique identifier in the event.

Particles GetBylds (std::vector< int > ids_)

Gets a vector of particles by their unique identifier in the event.

Particles GetByRole (int role_)

Gets a list of particles by their role in the event.

Particles GetDaughters (Particle *part_)

Gets a vector containing all the daughters from a particle.

• std::string GetLHERecord (const double weight_=1.)

Gets the LHE block for this event.

- Particle * GetMother (Particle *part_)
- Particle * GetOneByRole (int role_)
- Particles GetParticles ()

Gets a vector of particles in the event.

- std::vector< int > GetRoles ()
- Particles GetStableParticles ()

Gets a vector of stable particles in the event.

int NumParticles ()

Number of particles in the event.

Event & operator= (const Event &)

Copies all the relevant quantities from one Event object to another.

• void Store (std::ofstream *, double weight_=1.)

Data Fields

float time_cpu

Private Attributes

- ParticlesMap __part
- Particle * np

6.1.1 Detailed Description

Class containing all the information on the in- and outgoing particles' kinematics

6.1.2 Member Function Documentation

```
6.1.2.1 int Event::AddParticle ( Particle * part_, bool replace_ = false )
```

Sets the information on one particle in the process

Parameters

in	part_	The Particle object to insert or modify in the event
in	replace_	Do we replace the particle if already present in the event or do we append
		another particle with the same role ?

Returns

- 1 if a new Particle object has been inserted in the event
- 0 if an existing Particle object has been modified
- -1 if the requested role to edit is undefined or incorrect

6.1.2.2 void Event::Dump (bool stable_ = false)

Dumps all the known information on every Particle object contained in this Event container in the output stream

Parameters

in	stable	Do we only show the stable particles in this event?

6.1.2.3 **Particle*** Event::GetByld (int id_)

Returns the pointer to the Particle object corresponding to a unique identifier in the event

Parameters

in	id	The unique identifier to this particle in the event
111	<i>1</i> 4	The unique identifier to this particle in the event

Returns

A pointer to the requested Particle object

6.1.2.4 Particles Event::GetBylds (std::vector< int > ids_) [inline]

Returns the pointers to the Particle objects corresponding to the unique identifiers in the event

Parameters

	• 1 -	h Tha and an aitheacht an an abhan an Palacara barada an bhaile a barada an air an an an
ı ın	ıds	The unique identifiers to the particles to be selected in the event
		The ample recitions to the particles to be solded in the event

Returns

A vector of pointers to the requested Particle objects

6.1.2.5 Particles Event::GetByRole (int role_)

Returns the list of pointers to the Particle objects corresponding to a certain role in the process kinematics Parameters

in role_ The role the particles have to play in the process	-
---	---

Returns

A vector of pointers to the requested Particle objects

6.1.2.6 Particles Event::GetDaughters (Particle * part_) [inline]

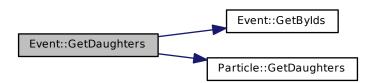
Parameters

in	part_	The particle for which the daughter particles have to be retrieved
----	-------	--

Returns

A Particle objects vector containing all the daughters' kinematic information

Here is the call graph for this function:



6.1.2.7 std::string Event::GetLHERecord (const double weight_ = 1.)

Returns an event block in a LHE format (a XML-style) with all the information on the particles composing this event

Parameters

in	weight_	The weight of the event	

Returns

A string containing the kinematic quantities for each of the particles in the event, formatted as the LHE standard requires.

6.1.2.8 Particle* Event::GetMother (Particle * part_) [inline]

Returns the pointer to the mother particle of any given Particle object in this event

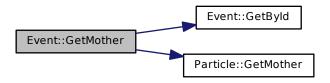
Parameters

in	part_	The pointer	to the	Particle	object	from	which	we	want	to	extract	the
		mother partic	cle									

Returns

A pointer to the mother Particle object

Here is the call graph for this function:



6.1.2.9 Particle* Event::GetOneByRole (int role_) [inline]

Returns the first Particle object in the particles list whose role corresponds to the given argument Parameters

in	role_	The role the particle has to play in the event
----	-------	--

Returns

A Particle object corresponding to the first particle found in this event

Here is the call graph for this function:



6.1.2.10 Particles Event::GetParticles (

Returns

A vector containing all the pointers to the $\mbox{\sc Particle}$ objects contained in the event

6.1.2.11 std::vector<int> Event::GetRoles (

Gets a list of roles for the given event (really process-dependant for the central system)

Returns

A vector of integers corresponding to all the roles the particles can play in the event

6.1.2.12 Particles Event::GetStableParticles ()

Returns

A vector containing all the pointers to the stable Particle objects contained in the event

6.1.2.13 int Event::NumParticles () [inline]

Returns

The number of particles in the event, as an integer

6.1.2.14 void Event::Store (std::ofstream * , double weight_ = 1.)

Stores in a file (raw format) all the kinematics on the outgoing leptons

Parameters

in weight_ | The weight of the event

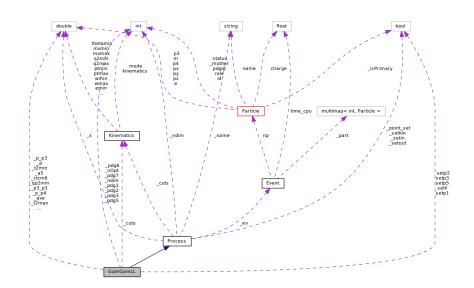
6.2 GamGamLL Class Reference

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

Inheritance diagram for GamGamLL:



Collaboration diagram for GamGamLL:



Public Member Functions

■ GamGamLL (int nOpt_=0)

Class constructor.

void ComputeCMenergy ()

Computes \sqrt{s} for the system.

double ComputeMX (double x_, double outmass_, double *dw_)

Computes the ougoing proton remnant mass.

double ComputeWeight ()

Computes the process' weight for the given point.

- virtual void **DumpPoint** ()
- void FillKinematics (bool)

Fills the Event object with the particles' kinematics.

- double GetD3 ()
- virtual Event * GetEvent ()

Returns the event content (list of particles with an assigned role)

- virtual std::string GetName ()
- double GetS1 ()
- double GetS2 ()
- double GetT1 ()
- void GetT1extrema (double &t1min_, double &t1max_)
- double GetT2 ()
- void GetT2extrema (double &t2min_, double &t2max_)
- double GetU1 ()
- double GetU2 ()
- double GetV1 ()
- double GetV2 ()
- virtual bool IsKinematicsDefined ()

Is the system's kinematics well defined?

- virtual unsigned int ndim () const
- void PrepareHadronisation (Particle *part_)
- bool SetIncomingParticles (Particle, Particle)

Sets the momentum and PDG id for the incoming particles.

void SetKinematics (Kinematics)

Sets the list of kinematic cuts to apply on the outgoing particles' final state.

bool SetOutgoingParticles (int, int)

Sets the PDG id for the outgoing particles.

virtual void SetPoint (const unsigned int ndim_, double x_[])

Sets the phase space point to compute.

- void StoreEvent (std::ofstream *, double)
- virtual double x (const unsigned int idx_)

Protected Attributes

Event * _ev

Event object containing all the information on the in- and outgoing particles.

- std::string __name
- bool _point_set

Is the phase space point set ?

bool _setin

Are the event's incoming particles set ?

bool _setkin

Is the full event's kinematic set ?

bool _setout

Are the event's outgoing particles set ?

Private Member Functions

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 ()

Private Attributes

- 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 	heta_3 of the first outgoing proton-like particle
  double <u>ct4</u>
      \cos 	heta_4 of the two-photons centre of mass system
  double <u>ct5</u>
      \cos 	heta_5 of the second outgoing proton-like particle
  double <u>ct6</u>
      \cos \theta_6 of the first outgoing lepton
double _ct7
      \cos\theta_7 of the second outgoing lepton
double <u>_ctcm6</u>
      \cos 	heta_6^{\rm CM} , production angle of the first outgoing lepton, computed in the centre of mass system.
Kinematics _cuts
  double _d1dq
  double _d1dq2
  double _d3
  double _dd1
  double _dd2
  double _dd3
  double _dd4
      \delta_5 = m_4^2 - t_1 as defined in Vermaseren's paper [4] for the full definition of this quantity
double _dd5
  double _de3
  double _de5
  double _delta
  double _dj
  double _dw31
 double _dw52
  double <u>e6lab</u>
       E_6^{
m lab}, energy of the first outgoing lepton, computed in the lab frame
 double <u>e7lab</u>
       E_7^{
m lab} , energy of the second outgoing lepton, computed in the lab frame
double <u>ec4</u>
      E_4, energy of the two-photon central system
double <u>eg1</u>
      Energy of the first central photon of momentum t_1.
double <u>eg2</u>
      Energy of the second central photon of momentum t_2.

 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 <u>ep2</u>
      E_2, energy of the second proton-like incoming particle
  double <u>ep3</u>
       E_3, energy of the first proton-like outgoing particle
```

double <u>ep5</u>

```
E_5, energy of the second proton-like outgoing particle
double <u>epsi</u>
 double _etot
      Total energy provided by the two incoming proton-like particles.
 double _g4
 double _g5
  double _g6
 double _gamma
      \gamma 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
unsigned 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_g1 [3]
      3-momentum of the second central photon of momentum t_1
double _p3_g2 [3]
      3-momentum of the second central photon of momentum t_{\mathrm{2}}
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
      PDG identifier of the first proton-like incoming particle.
int _pdg2
      PDG identifier of the second proton-like incoming particle.
int _pdg3
      PDG identifier of the first proton-like outgoing particle.

    int _pdg5

      PDG identifier of the second proton-like outgoing particle.
int _pdg6
      PDG identifier of the first outgoing lepton.

    int _pdg7

      PDG identifier of the second outgoing lepton.
■ 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 _plab_ip1 [4]

    double __plab__ip2 [4]

    double __plab__ol1 [4]

 double _plab_ol2 [4]

    double __plab__op1 [4]

  double _plab_op2 [4]

    double __plab__ph1 [4]

    double __plab__ph2 [4]

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
      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 <u>_st3</u>
      \sin \theta_3 of the first outgoing proton-like particle
double <u>st4</u>
      \sin 	heta_4 of the two-photons centre of mass system
double <u>_st5</u>
      \sin \theta_5 of the second outgoing proton-like particle
double <u>_st6</u>
      \sin \theta_6 of the first outgoing lepton
■ double st7
      \sin \theta_7 of the second outgoing lepton
 double stcm6
      \sin \theta_6^{\rm CM}, production angle of the first outgoing lepton, computed in the centre of mass system.
 double _{f t1}
double <u>_t1max</u>
double _t1min
 double _t2
double <u>_t2max</u>
 double _t2min
double <u>tau</u>
      \delta_6=m_4^2-m_5^2 as defined in Vermaseren's paper [4] for the full definition of this quantity
double _u1
double _u2
double _v1
double _v2
 double <u>w</u>1
      m_1^2, squared mass of the first proton-like incoming particle
 double <u>w12</u>
      \delta_2=m_1^2-m_2^2 as defined in Vermaseren's paper [4] for the full definition of this quantity
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

 $\delta_1=m_3^2-m_1^2$ as defined in Vermaseren's paper [4] for the full definition of this quantity

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

 $\delta_4=m_5^2-m_2^2$ as defined in Vermaseren's paper [4] for the full definition of this quantity

■ 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 * _x

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

bool setll

Is the outgoing leptons' state 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 ?

6.2.1 Detailed Description

Full class of methods and objects to compute the full analytic matrix element [4] for the $\gamma\gamma \to \ell^+\ell^-$ process according to a set of kinematic constraints provided for the incoming and outgoing particles (the Kinematics object). The particle roles in this process are defined as following :

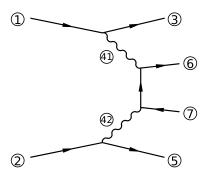


Figure 1: Detailed particle roles in the two-photon

process as defined by the *GamGamLL* object. The incoming protons/electrons are denoted by a role 1, and 2, as the outgoing protons/protons remnants/ electrons carry the indices 3 and 5. The two outgoing leptons have the roles 6 and 7, while the lepton/antilepton distinction is done randomly (thus, the arrow convention is irrelevant here).

- 6.2.2 Constructor & Destructor Documentation
- $6.2.2.1 \quad \mathsf{Gam} \mathsf{GamLL} :: \mathsf{Gam} \mathsf{GamLL} \ (\ \, \mathsf{int} \, \, \mathsf{nOpt} \underline{\ \ } = 0 \ \,)$

Sets the mandatory parameters used in the methods computing the kinematics and the cross-section for the $\gamma\gamma\to\ell^+\ell^-$ process

Parameters

in	nOpt_	Optimisation???

Todo Figure out how this *nOpt*_ parameter is affecting the final cross-section computation and events generation

6.2.3 Member Function Documentation

6.2.3.1 void GamGamLL::ComputeCMenergy ()

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

6.2.3.2 double GamGamLL::ComputeMX (double $x_{,}$ double outmass_, double $* dw_{,}$)

Computes the mass of the outgoing proton remnant if any

Parameters

in	x_	A random number (between 0 and 1)
in	outmass_	The maximal outgoing particles' invariant mass
out	dw_	The size of the integration bin

Returns

The mass of the outgoing proton remnant

6.2.3.3 double GamGamLL::ComputeWeight () [virtual]

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

Parameters

in	nm_	???

Returns

 $\frac{d\sigma}{dx}(\gamma\gamma\to\ell^+\ell^-)$, the differential cross-section for the given point in the phase space.

Todo Find out what this *nm*_ parameter does...

Implements Process.

6.2.3.4 void GamGamLL::FillKinematics (bool symmetrise_) [virtual]

Fills the private Event object with all the Particle object contained in this event.

Parameters

in	symmetrise_	Do we have to symmetrise the event (randomise the production of the
		positively- and negatively-charged lepton) ?

Reimplemented from Process.

6.2.3.5 virtual **Event*** Process::GetEvent () [inline], [virtual], [inherited]

Returns the complete list of Particle with their role in the process for the point considered in the phase space as an Event object.

Returns

The Event object containing all the generated Particle objects

6.2.3.6 double GamGamLL::GetT1 () [inline]

Returns the value for the first photon virtuality

Returns

 t_1 , the first photon virtuality

6.2.3.7 void GamGamLL::GetT1extrema (double & t1min_, double & t1max_) [inline]

Returns the two limit values for the first photon virtuality

Parameters

out	t1min_	The minimal value for t_1
out	t1max_	The maximal value for t_1

6.2.3.8 double GamGamLL::GetT2 () [inline]

Returns the value for the second photon virtuality

Returns

 t_2 , the second photon virtuality

6.2.3.9 void GamGamLL::GetT2extrema (double & t2min_, double & t2max_) [inline]

Returns the two limit values for the second photon virtuality

Parameters

out	t2min_	The minimal value for t_2
out	t2max_	The maximal value for t_2

6.2.3.10 virtual bool Process::lsKinematicsDefined () [inline], [virtual], [inherited]

Is the system's kinematics well defined and compatible with the process? This check is mandatory to perform the (_ndim)-dimensional point's cross-section computation.

Returns

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

6.2.3.11 bool GamGamLL::Orient () [private]

Calculates energies and momenta of the 1st, 2nd (resp. the "proton-like" and the "electron-like" incoming particles), 3rd (the "proton-like" outgoing particle), 4th (the two-photons central system) and 5th (the "electron-like" outgoing particle) particles in the overall centre of mass frame.

6.2.3.12 double GamGamLL::PeriPP (int , int) [private]

Contains the expression of the matrix element squared for the $\gamma\gamma \to \ell^+\ell^-$ process. It returns the value of the convolution of the form factor or structure functions with the central two-photons matrix element squared.

Returns

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

```
6.2.3.13 bool GamGamLL::Pickin ( ) [private]
```

Describes the kinematics of the process $p_1 + p_2 \rightarrow p_3 + p_4 + p_5$ in terms of Lorentz-invariant variables. These variables (along with others) will then be feeded into the *PeriPP* method (thus are essential for the evaluation of the full matrix element).

```
6.2.3.14 void GamGamLL::PrepareHadronisation ( Particle * part_ )
```

Sets all the kinematic variables for the outgoing proton remnants in order to be able to hadronise them afterwards

Parameters

in	part_	Particle to "prepare" for the hadronisation to be performed	

```
6.2.3.15 bool GamGamLL::SetIncomingParticles ( Particle ip1_, Particle ip2_ ) [virtual]
```

Specifies the incoming particles' kinematics as well as their properties using two Particle objects.

Parameters

in	ip1	Information on the first incoming particle
in	ip2_	Information on the second incoming particle

Returns

A boolean stating whether or not the incoming kinematics is properly set for this event

Reimplemented from Process.

```
6.2.3.16 void GamGamLL::SetKinematics ( Kinematics cuts_ ) [virtual]
```

Parameters

in	n cuts_ The Cuts object containing the kinematic parameters
----	---

Reimplemented from Process.

```
6.2.3.17 bool GamGamLL::SetOutgoingParticles ( int part_, int pdgld_ ) [virtual]
```

Parameters

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

Returns

A boolean stating whether or not the outgoing kinematics is properly set for this event

Reimplemented from Process.

Sets the phase space point to compute the weight associated to it.

Parameters

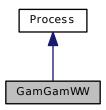
Generated on Wed Apr 16 2014 22:06:02 for LPAIR++ by Doxygen

in	ndim_	The number of dimensions of the point in the phase space
in	x_[]	The (ndim_)-dimensional point in the phase space on which the kinematics
		and the cross-section are computed

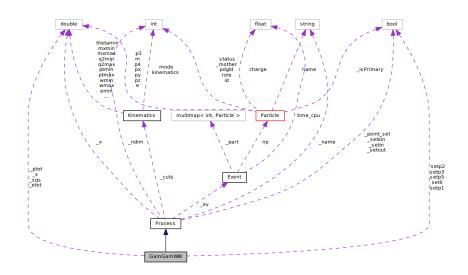
6.3 GamGamWW Class Reference

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

Inheritance diagram for GamGamWW:



Collaboration diagram for GamGamWW:



Public Member Functions

- void ComputeCMenergy ()
- double ComputeMX (double x_, double outmass_, double *dw_)
- double ComputeWeight ()

Returns the weight for this point in the phase-space.

- virtual void DumpPoint ()
- void FillKinematics (bool)

Fills the Event object with the particles' kinematics.

```
virtual Event * GetEvent ()
          Returns the event content (list of particles with an assigned role)
   virtual std::string GetName ()

    virtual bool IsKinematicsDefined ()

          Is the system's kinematics well defined?
   • virtual unsigned int ndim () const
     void PrepareHadronisation (Particle *part )

    bool SetIncomingParticles (Particle, Particle)

         Sets the momentum and PDG id for the incoming particles.
   void SetKinematics (Kinematics)
          Sets the list of kinematic cuts to apply on the outgoing particles' final state.

    bool SetOutgoingParticles (int, int)

         Sets the PDG id for the outgoing particles.

    virtual void SetPoint (const unsigned int ndim_, double x_[])

         Sets the phase space point to compute.
   void StoreEvent (std::ofstream *, double)
      virtual double x (const unsigned int idx_)
Protected Attributes
   Kinematics _cuts
         Set of cuts to apply on the final phase space.

    Event * _ev

          Event object containing all the information on the in- and outgoing particles.
   std::string __name
     unsigned int _ndim
   bool _point_set
         Is the phase space point set ?
   bool _setin
         Are the event's incoming particles set ?
   bool _setkin
         Is the full event's kinematic set ?
   bool _setout
         Are the event's outgoing particles set ?

    double * _x

Private Attributes
   double etot
          Total energy provided by the two incoming proton-like particles.
     double _ptot
          Total momentum provided by the two incoming proton-like particles (along the z-axis)
   double _s
         s, squared centre of mass energy of the incoming particles' system
   double _sqs
          \sqrt{s}, centre of mass energy of the incoming particles' system
   bool setll
         Is the outgoing leptons' state 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?

- 6.3.1 Member Function Documentation
- 6.3.1.1 void GamGamWW::FillKinematics (bool symmetrise_) [virtual]

Fills the private Event object with all the Particle object contained in this event.

Parameters

in	symmetrise_	Do we have to symmetrise the event (randomise the production of the
		positively- and negatively-charged lepton) ?

Reimplemented from Process.

```
6.3.1.2 virtual Event* Process::GetEvent ( ) [inline], [virtual], [inherited]
```

Returns the complete list of Particle with their role in the process for the point considered in the phase space as an Event object.

Returns

The Event object containing all the generated Particle objects

```
6.3.1.3 virtual bool Process::IsKinematicsDefined ( ) [inline], [virtual], [inherited]
```

Is the system's kinematics well defined and compatible with the process? This check is mandatory to perform the (_ndim)-dimensional point's cross-section computation.

Returns

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

6.3.1.4 bool GamGamWW::SetIncomingParticles (Particle ip1_, Particle ip2_) [virtual]

Specifies the incoming particles' kinematics as well as their properties using two Particle objects.

Parameters

in	ip1	Information on the first incoming particle
in	ip2_	Information on the second incoming particle

Returns

A boolean stating whether or not the incoming kinematics is properly set for this event

Reimplemented from Process.

6.3.1.5 void GamGamWW::SetKinematics (Kinematics cuts_) [virtual]

Parameters

Γ	in	cuts	The Cuts object containing the kinematic parameters

Reimplemented from Process.

6.3.1.6 bool GamGamWW::SetOutgoingParticles (int part_, int pdgld_) [virtual]

Parameters

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

Returns

A boolean stating whether or not the outgoing kinematics is properly set for this event

Reimplemented from Process.

6.3.1.7 virtual void Process::SetPoint (const unsigned int ndim_, double x_[]) [virtual], [inherited]

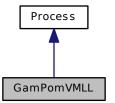
Sets the phase space point to compute the weight associated to it.

Parameters

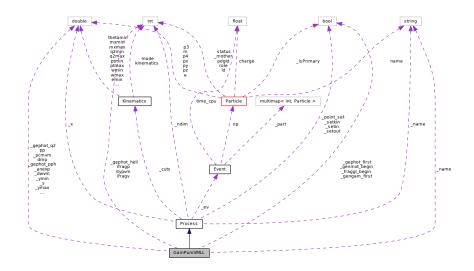
in	ndim_	The number of dimensions of the point in the phase space
in	x_[]	The (ndim_)-dimensional point in the phase space on which the kinematics
		and the cross-section are computed

6.4 GamPomVMLL Class Reference

Computes the matrix element for a CE $\gamma \mathbb{P} \to J/\psi, \Upsilon \to \ell^+\ell^-$ process. Inheritance diagram for GamPomVMLL:



Collaboration diagram for GamPomVMLL:



Public Member Functions

- void ComputeCMenergy ()
- double ComputeMX (double x_, double outmass_, double *dw_)
- double ComputeWeight ()

Returns the weight for this point in the phase-space.

- virtual void DumpPoint ()
- void FillKinematics (bool)

Fills the Event object with the particles' kinematics.

virtual Event * GetEvent ()

Returns the event content (list of particles with an assigned role)

- virtual std::string GetName ()
- virtual bool lsKinematicsDefined ()

Is the system's kinematics well defined?

- virtual unsigned int ndim () const
- void PrepareHadronisation (Particle *part_)
- bool SetIncomingParticles (Particle, Particle)

Sets the momentum and PDG id for the incoming particles.

void SetKinematics (Kinematics)

Sets the list of kinematic cuts to apply on the outgoing particles' final state.

bool SetOutgoingParticles (int, int)

Sets the PDG id for the outgoing particles.

virtual void SetPoint (const unsigned int ndim_, double x_[])

Sets the phase space point to compute.

- void StoreEvent (std::ofstream *, double)
- virtual double x (const unsigned int idx_)

Protected Member Functions

- void DecVM ()
- void FragGI ()
- void GenGam ()

```
    void GEPhot (int igammd_=1)
    double PXMass (double mmin_, double mmax_)
```

double VXMass (double mmin_, double mmax_)

Protected Attributes

Kinematics _cuts

Set of cuts to apply on the final phase space.

■ Event * ev

Event object containing all the information on the in- and outgoing particles.

- unsigned int _ndim
- bool _point_set

Is the phase space point set ?

bool _setin

Are the event's incoming particles set ?

bool _setkin

Is the full event's kinematic set ?

bool _setout

Are the event's outgoing particles set ?

double * _x

Private Attributes

double _alpha1

Slope α' of pomeron trajectory in GeV^{-2} .

double <u>_amxb0</u>

Mass of diffractively dissociating hadronic system for which _b0 was measured.

- double _anexp
- double _b0
- double _dmvm

Mass of generated vector meson.

double _dwvm

Width of generated vector meson.

- double <u>ecm</u>
- double _epsilm

Intercept of pomeron trajectory minus 1.

double _epsilw

Intercept of pomeron trajectory minus 1.

- bool __fraggl__begin
- bool _gengam_first
- bool <u>_genmxt_begin</u>
- bool <u>gephot_first</u>
- int **_gephot_heli**
- double <u>gephot_pel</u> [5]
- double _gephot_ppe [5]
- double _gephot_pph [5]
- double <u>gephot</u> [5]
- double <u>gephot</u>q2
- std::string __name
- double _pcm3

CM momentum of outgoing particles.

double _pcmvm [3]

```
double _ppcms8 [1000][5]
     double _q2
         Absolute of square-momentum of virtual photon.
   double _q2max
         Maximal Q^2 of photon in GeV^2.
   double _q2min
         Minimal Q^2 of photon in {
m GeV}^2.
   double _s
   double _w2
     double <u>wb0</u>
         CM energy of \gamma p system at which _b0 was measured, in GeV.
   double _wmax
         Maximal CM energy of \gamma p system.
   double _wmin
         Minimal CM energy of \gamma p system.
   double _wsig0
         \gamma p CM energy at which SIGGP was measured
   double _ymax
         Maximal value of scaling variable y.
   double _ymin
         Minimal value of scaling variable y.
   double dme
   double dmp
   int ifragp
   int ifragv
   int itypvm
         PDG code for produced vector meson.
   double pe
   double pp
6.4.1 Member Function Documentation
6.4.1.1 void GamPomVMLL::DecVM ( ) [protected]
Let the generated vector meson decay
Author
     Benno List
Date
     25 jan 1993
6.4.1.2 void GamPomVMLL::FillKinematics ( bool symmetrise_ ) [virtual]
Fills the private Event object with all the Particle object contained in this event.
Parameters
```

in	symmetrise_	Do we have to symmetrise the event (randomise the production of the
		positively- and negatively-charged lepton) ?

Reimplemented from Process.

6.4.1.3 void GamPomVMLL::GEPhot (int igammd_ = 1) [protected]

Generate one event with unweighted photon & electron

- according to WWA :
 - transversal photonspectrum. $Q^2 \rightarrow 0$:

$$P(y,Q^2) = \tfrac{\alpha}{2\pi} \tfrac{1}{Q^2 y} \left(2(1-y) \left(1 - \tfrac{Q^2_{\min}}{Q^2} \right) + y^2 \right)$$

– longitudinal photonspectrum. $Q^2 \rightarrow 0$:

$$P(y,Q^2) = \frac{\alpha}{2\pi} \frac{1}{Q^2 y} \left(2(1-y)\right)$$

- full transversal photonspectrum given by:
 - ABT, I. & J.R. SMITH (1992): MC upgrades to study untagged events. H1-10/92-249.
 - SMITH, J.R. (1992): An experimentalist's guide to photon flux calculations. H1-12/92-259
 - SMITH, J.R. & B.D. BUROW (1994): Photon fluxes with beam mass effects and polarizations. -H1-01/94-338.
- full transversal and longitudinal spectrum by ABT&SMITH
 - calculate integrated factor over the spectrum: kinematical bounds : $[Y_{\min}, Y_{\max}](W_{\min})$, $[Q^2_{\min}, Q^2_{\max}](Q^2_{\mathrm{cutoff}})$

igammd_ Photon * 1: V

Photon generation mode:

- * 1: WWA/EPA approximation (including electron-mass effect and longitudinal flux). **Recommended**
- * 2: Transverse spectrum
- * 3: Transverse & longitudinal spectrum
- * 4: as 3, but flux in proton rest frame

6.4.1.4 virtual **Event*** Process::GetEvent () [inline], [virtual], [inherited]

Returns the complete list of Particle with their role in the process for the point considered in the phase space as an Event object.

Returns

The Event object containing all the generated Particle objects

6.4.1.5 virtual bool Process::IsKinematicsDefined () [inline], [virtual], [inherited]

Is the system's kinematics well defined and compatible with the process? This check is mandatory to perform the (*ndim*)-dimensional point's cross-section computation.

Returns

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

6.4.1.6 bool GamPomVMLL::SetIncomingParticles (Particle ip1_, Particle ip2_) [virtual]

Specifies the incoming particles' kinematics as well as their properties using two Particle objects.

Parameters

in	ip1	Information on the first incoming particle
in	ip2_	Information on the second incoming particle

Returns

A boolean stating whether or not the incoming kinematics is properly set for this event

Reimplemented from Process.

6.4.1.7 void GamPomVMLL::SetKinematics (Kinematics cuts_) [virtual]

Parameters

in	cuts_	The Cuts object containing the kinematic parameters

Reimplemented from Process.

6.4.1.8 bool GamPomVMLL::SetOutgoingParticles (int part_, int pdgld_) [virtual]

Parameters

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

Returns

A boolean stating whether or not the outgoing kinematics is properly set for this event

Reimplemented from Process.

6.4.1.9 virtual void Process::SetPoint (const unsigned int ndim_, double x_[]) [virtual], [inherited]

Sets the phase space point to compute the weight associated to it.

Parameters

in	ndim_	The number of dimensions of the point in the phase space
in	x_[]	The (ndim_)-dimensional point in the phase space on which the kinematics
		and the cross-section are computed

6.4.1.10 double GamPomVMLL::VXMass (double mmin_, double mmax_) [protected]

Generate hadronic mass between mmin_ and mmax_ for VM vertex

Parameters

mmin_	Minimal allowed mass
mmax_	Maximal allowed mass

Returns

Hadronic mass in GeV

Author

Benno List

Date

14 jan 1992

6.4.2 Field Documentation

6.4.2.1 double GamPomVMLL::_alpha1 [private]

Note

Controls shrinkage of b slope

6.4.2.2 double GamPomVMLL:: amxb0 [private]

Note

For _amxb0=0.0, _amxb0 is set according to production mode. Value is not meaningful for elastic VM production

6.4.2.3 double GamPomVMLL::_anexp [private]

Power law exponent.

- For $_anexp=0$ (default), a pure exponential spectrum is generated according to (taking t<0) $\frac{\mathrm{d}\sigma}{\mathrm{d}t}=e^{bt}$
- For $_anexp > 1$, an interpolated spectrum is generated according to $\frac{\mathrm{d}\sigma}{\mathrm{d}t} = \exp\left[-n\ln\left(-\frac{bt}{n}+1\right)\right] = \left(-\frac{bt}{n}+1\right)^{-n}$ with $n=_anexp$
 - Limit for small bt: $\exp\left(bt+ct^2\right)$ with $c=b^2/2n$
 - Limit for large $bt \gg n$: t^{-n}
- 6.4.2.4 double GamPomVMLL::_b0 [private]

Slope parameter b of t distribution in ${
m GeV}^{-2}$ at CM energy $_wb0$ and (for diffractive dissociation) mass $_amxb0$

Note

Must be positive!

6.4.2.5 double GamPomVMLL::_epsilm [private]

Note

Controls M_X spectrum

6.4.2.6 double GamPomVMLL::_epsilw [private]

Note

Controls rise of $\sigma_{\gamma p}$ with W

6.4.2.7 double GamPomVMLL::_wmax [private]

Note

If too low, it is set to \sqrt{s}

6.4.2.8 int GamPomVMLL::itypvm [private]

PDG code for produced vector meson (should have $J^{PC}=1^{--}$) Possible values:

113 : ρ

■ 223 : ω

■ 333 : *φ*

• 443 : J/ψ

■ 20443 : ψ′

■ 553 : $\Upsilon(1s)$

■ 20553 : $\Upsilon(2s)$

■ $30553 : \Upsilon(3s)$

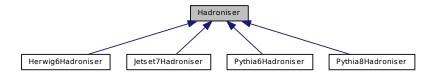
• 40113 : $\rho(1450) \to \pi^+ \pi^- \rho^0$

• 10333 : $\phi(1680) \to K\bar{K}$

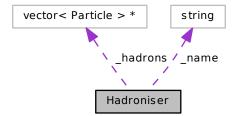
• 22 : diffr. gamma dissoc. (special value)

6.5 Hadroniser Class Reference

Inheritance diagram for Hadroniser:



Collaboration diagram for Hadroniser:



Public Member Functions

- std::vector< Particle > GetHadrons ()
- std::string GetName ()
- virtual bool Hadronise (Particle *part_)

Main caller to hadronise a particle.

virtual bool Hadronise (Event *ev_)

Hadronises a full event.

Protected Attributes

std::vector< Particle > * _hadrons

List of hadrons produced by this hadronisation process.

std::string __name

Name of the hadroniser.

6.5.1 Detailed Description

Class template to define any hadroniser as a general object with defined methods

Author

Laurent Forthomme laurent.forthomme@uclouvain.be

Date

January 2014

6.5.2 Member Function Documentation

6.5.2.1 std::vector<**Particle**> Hadroniser::GetHadrons () [inline]

Gets the full list of hadrons (as Particle objects) produced by the hadronisation

Returns

A vector of Particle containing all the hadrons produced

6.5.2.2 virtual bool Hadroniser::Hadronise (**Event** * ev_) [inline], [virtual]

Launches the hadroniser on the full event information

Parameters

in,o	ut ev_	The event to hadronise

Returns

A boolean stating whether or not the hadronisation occured successfully

Reimplemented in Pythia6Hadroniser, Herwig6Hadroniser, Jetset7Hadroniser, and Pythia8Hadroniser.

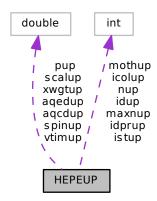
Here is the call graph for this function:



6.6 HEPEUP Class Reference

User-process event information.

Collaboration diagram for HEPEUP:



Public Member Functions

• **HEPEUP** (const int nup_=500)

Data Fields

double aqcdup

QCD coupling $\alpha_{\rm QCD}$ used for this event.

double aqedup

QED coupling $\alpha_{\rm QED}$ used for this event.

■ int * icolup [2]

Index for the colour flow line passing through the colour (resp. anti-colour) of the particle.

• int idprup

ID of the process in this event.

■ int * idup

Particle ID according to the Particle Data Group convention.

■ int * istup

Status code.

■ int * mothup [2]

Index of first and last mother.

■ int nup

Number of particle entries in this event.

double * pup [5]

Lab-frame momentum of the particle, in GeV.

double scalup

Scale of the event in GeV, as used for the calculation of PDFs.

double * spinup

Cosine of the angle between the spin-vector of the particle and the 3-momentum of the decaying particle, in the lab frame.

double * vtimup

Invariant lifetime c au in mm.

double xwgtup

Event weight.

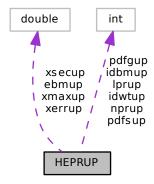
Static Public Attributes

static const int maxnup = 500
 Maximum number of particle entries.

6.7 HEPRUP Class Reference

Generic user-process interface for events generator.

Collaboration diagram for HEPRUP:



Public Member Functions

■ **HEPRUP** (const int nprup_=1)

Data Fields

double ebmup [2]

Energy in GeV of the beam 1 and 2 particles.

• int idbmup [2]

ID of the beam 1 and 2 particles according to the Particle Data Group convention.

- int idwtup
- int * **Iprup**
- int nprup
- int pdfgup [2]

Author group for beam 1 and 2, according to PDFLIB.

• int pdfsup [2]

PDF set ID for beam 1 and 2, according to PDFLIB.

- double * xerrup
- double * xmaxup
- double * xsecup

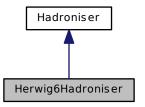
6.7.1 Detailed Description

User-process run information

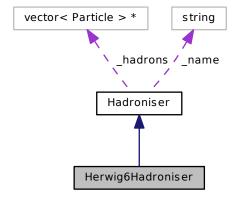
6.8 Herwig6Hadroniser Class Reference

Herwig6 hadronisation algorithm.

Inheritance diagram for Herwig6Hadroniser:



Collaboration diagram for Herwig6Hadroniser:



Public Member Functions

- std::vector< Particle > GetHadrons ()
- std::string GetName ()
- virtual bool Hadronise (Particle *part_)

Main caller to hadronise a particle.

bool Hadronise (Event *ev_)

Hadronises a full event.

Protected Attributes

std::vector< Particle > * _hadrons

List of hadrons produced by this hadronisation process.

std::string __name

Name of the hadroniser.

Static Private Member Functions

static void hwdhad ()

6.8.1 Member Function Documentation

6.8.1.1 std::vector<**Particle**> Hadroniser::GetHadrons() [inline], [inherited]

Gets the full list of hadrons (as Particle objects) produced by the hadronisation

Returns

A vector of Particle containing all the hadrons produced

6.8.1.2 bool Herwig6Hadroniser::Hadronise (**Event** * ev_) [virtual]

Launches the hadroniser on the full event information

Parameters

in,out	ev_	The event to hadronise
--------	-----	------------------------

Returns

A boolean stating whether or not the hadronisation occured successfully

Reimplemented from Hadroniser.

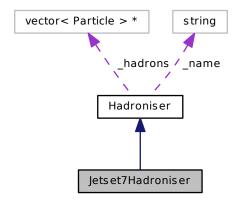
6.9 Jetset7Hadroniser Class Reference

Jetset7 hadronisation algorithm.

Inheritance diagram for Jetset7Hadroniser:



Collaboration diagram for Jetset7Hadroniser:



Public Member Functions

- std::vector< Particle > GetHadrons ()
- std::string GetName ()
- bool Hadronise (Particle *part_)

Main caller to hadronise a particle.

bool Hadronise (Event *ev_)

Hadronises a full event.

Protected Attributes

std::vector< Particle > * _hadrons

List of hadrons produced by this hadronisation process.

std::string __name

Name of the hadroniser.

Private Member Functions

bool PrepareHadronisation (Event *ev_)

Static Private Member Functions

- static float luchge (int pdgid_)
- static void luexec ()
- static void lugive (const std::string &line_)
- static void lujoin (int njoin_, int ijoin_[2])

Connect entries with colour flow information.

- static void lulist (int mlist_)
- static std::string luname (int pdgid_)
- static double ulmass (int pdgid_)

6.9.1 Member Function Documentation

6.9.1.1 std::vector< Particle> Hadroniser::GetHadrons () [inline], [inherited]

Gets the full list of hadrons (as Particle objects) produced by the hadronisation

Returns

A vector of Particle containing all the hadrons produced

6.9.1.2 bool Jetset7Hadroniser::Hadronise (**Event** * ev_) [virtual]

Launches the hadroniser on the full event information

Parameters

in,out	ev_	The event to hadronise
--------	-----	------------------------

Returns

A boolean stating whether or not the hadronisation occured successfully

Reimplemented from Hadroniser.

6.9.1.3 static void Jetset7Hadroniser::lujoin (int njoin_, int ijoin_[2]) [inline], [static], [private]

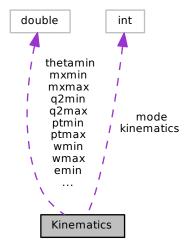
Parameters

njoin_	Number of particles to join in the colour flow
ijoin_	List of particles to join in the colour flow

6.10 Kinematics Class Reference

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

Collaboration diagram for Kinematics:



Public Member Functions

void Dump ()

Dumps all the parameters used in this process cross-section computation / events generation.

Data Fields

double emax

Maximal energy of the central two-photons system.

double emin

Minimal energy of the central two-photons system.

int kinematics

Type of kinematics to consider for the phase space.

• int mode

Sets of cuts to apply on the final phase space.

double mxmax

Maximal mass (in GeV/c 2) of the outgoing proton remnant(s)

■ double mxmin

Minimal mass (in GeV/c^2) of the outgoing proton remnant(s)

double ptmax

Maximal transverse momentum of the single outgoing leptons.

double ptmin

Minimal transverse momentum of the single outgoing leptons.

double q2max

The maximal value of Q^2 .

double q2min

The minimal value of Q^2 .

double thetamax

Maximal polar ($\theta_{\rm max}$) angle of the outgoing leptons, expressed in degrees.

double thetamin

Minimal polar (θ_{\min}) angle of the outgoing leptons, expressed in degrees.

double wmax

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.

double wmin

The minimal s on which the cross section is integrated.

6.10.1 Field Documentation

6.10.1.1 int Kinematics::kinematics

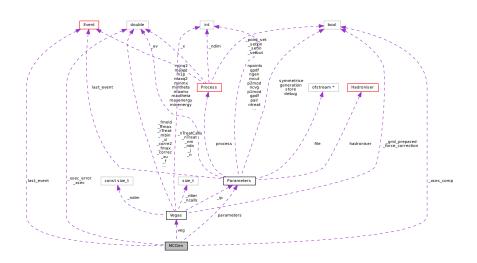
Type of kinematics to consider for the process. Can either be :

- 0 for the electron-electron elastic case
- 1 for the proton-proton elastic case
- 2 for the proton-proton single-dissociative (or inelastic) case
- 3 for the proton-proton double-dissociative case

6.11 MCGen Class Reference

Core of the Monte-Carlo generator.

Collaboration diagram for MCGen:



Public Member Functions

MCGen ()

Class constructor.

MCGen (Parameters *ip_)

Class constructor.

void AnalyzePhaseSpace (const std::string)

Returns the set of parameters used to setup the phase space to integrate.

- void ComputeXsection (double *xsec_, double *err_)
 - Compute the cross-section for the given process.
- Event * GenerateOneEvent ()
- void LaunchGeneration ()
- void PrintHeader ()
- void Test ()

Data Fields

Event * last_event

Last event generated in this run.

Parameters * parameters

Private Member Functions

void BuildVegas ()

Private Attributes

- double _xsec
- bool _xsec_comp
- double _xsec_error

■ Vegas * veg

The Vegas integrator which will integrate the function.

6.11.1 Detailed Description

This object represents the core of this Monte Carlo generator, with its allowance to generate the events (using the embedded Vegas object) and to study the phase space in term of the variation of resulting cross section while scanning the various parameters (point x in the DIM-dimensional phase space).

The phase space is constrained using the InputParameters object given as an argument to the constructor, and the differential cross-sections for each value of the array \mathbf{x} are computed in the f-function defined outside (but populated inside) this object.

This f-function embeds a Process object which defines all the methods to obtain this differential cross-section as well as the in- and outgoing kinematics associated to each particle.

Author

Laurent Forthomme laurent.forthomme@uclouvain.be

Date

February 2013

6.11.2 Constructor & Destructor Documentation

6.11.2.1 MCGen::MCGen ()

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

```
6.11.2.2 MCGen::MCGen ( Parameters * 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

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

6.11.3 Member Function Documentation

6.11.3.1 void MCGen::AnalyzePhaseSpace (const std::string)

Returns

The Parameter object embedded in this class

6.11.3.2 void MCGen::ComputeXsection (double * xsec_, double * err_)

Computes the cross-section for the run defined by this object. This returns the cross-section as well as the absolute error computed along.

Parameters

out	xsec_	The computed cross-section, in pb
out	err_	The absolute integration error on the computed cross-section, in pb

6.11.3.3 **Event*** MCGen::GenerateOneEvent ()

Generates one single event given the phase space computed by Vegas in the integration step

Returns

A pointer to the **Event** object generated in this run

6.11.3.4 void MCGen::LaunchGeneration ()

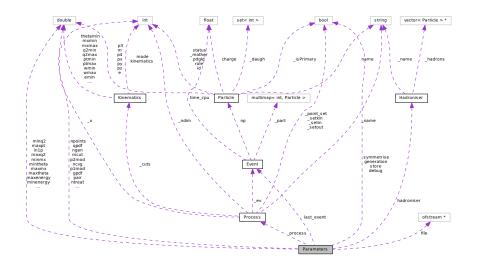
Launches the full events generation

Deprecated This method is to be suppressed since the events generation can now be launched one event at a time using the *GenerateOneEvent* method

6.12 Parameters Class Reference

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

Collaboration diagram for Parameters:



Public Member Functions

void Dump ()

Dumps the input parameters in the console.

bool ReadConfigFile (std::string inFile_)

Reads content from config file to load the variables.

void SetEtaRange (double etamin_, double etamax_)

Sets the pseudo-rapidity range for the produced leptons.

bool StoreConfigFile (std::string outFile_)

Stores the full run configuration to an external config file.

Data Fields

bool debug

Do we need control plots all along the process?

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)

• int gpdf

PDFLIB group to use.

■ Hadroniser * hadroniser

Hadronisation algorithm to use for the proton(s) remnants fragmentation.

double in1p

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

double in2p

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

int itvg

Maximal number of iterations to perform by VEGAS.

Event * last_event

The pointer to the last event produced in this run.

double maxenergy

Maximal energy of the outgoing leptons.

int maxgen

Maximal number of events to generate in this run.

double maxmx

Maximal M_X of the outgoing proton remnants.

double maxpt

Maximal p_T of the outgoing leptons.

double maxq2

Maximal value of Q^2 , the internal photons lines' virtuality.

double maxtheta

Maximal polar angle θ of the outgoing leptons.

int mcut

Set of cuts to apply on the outgoing leptons.

double minenergy

Minimal energy of the outgoing leptons.

double minmx

Minimal M_X of the outgoing proton remnants.

double minpt

Minimal p_T of the outgoing leptons.

double minq2

Minimal value of Q^2 , the internal photons lines' virtuality.

double mintheta

Minimal polar angle θ of the outgoing leptons.

- int ncvg
- int ngen

Number of events already generated in this run.

int npoints

Number of points to "shoot" in each integration bin by the algorithm.

int ntreat

Maximal number of TREAT calls.

• int p1mod

First particle's mode.

int p2mod

Second particle's mode.

• int pair

PDG id of the outgoing leptons.

- Process * process
- int qpdf

Number of quarks.

• int spdf

PDFLIB set to use.

bool store

Are the events generated in this run to be stored in the output file?

bool symmetrise

Control plots objects.

6.12.1 Detailed Description

Note

The default parameters are derived from GMUINI in LPAIR

6.12.2 Member Function Documentation

6.12.2.1 bool Parameters::ReadConfigFile (std::string inFile_)

Reads the list of parameters to be used in this cross-section computation/events generation from an external input card.

Parameters

in	inFile_	Name of the configuration file to load
----	---------	--

Returns

A boolean stating whether this input configuration file is correct or not

6.12.2.2 void Parameters::SetEtaRange (double etamin_, double etamax_)

Defines the range to cover in pseudo-rapidity for the outgoing leptons produced in this process. This method converts this range into a range in θ , the polar angle.

Parameters

in	etamin_	The minimal value of η for the outgoing leptons
in	etamax_	The maximal value of η for the outgoing leptons

6.12.2.3 bool Parameters::StoreConfigFile (std::string outFile_)

Parameters

in	outFile_	Name of the configuration file to create

Returns

A boolean stating whether this output configuration file is correctly written or not

6.12.3 Field Documentation

6.12.3.1 bool Parameters::debug

Enables or disables the production of control plots for several kinematic quantities in this process

6.12.3.2 double Parameters::maxmx

Maximal mass of the outgoing proton remnants, M_X , in GeV/c 2 .

6.12.3.3 double Parameters::maxpt

Maximal transverse momentum cut to apply on the outgoing lepton(s)

6.12.3.4 int Parameters::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/c, or

- 0.75
$$< rac{|p_z|}{|\mathbf{p}|} \le$$
 0.95 and $p_z > 1$ GeV/c,

- 2 Cuts on both the outgoing leptons, according to the provided cuts parameters
- 3 Cuts on at least one outgoing lepton, according to the provided cut parameters

6.12.3.5 double Parameters::minmx

Minimal mass of the outgoing proton remnants, M_X , in GeV/c 2 .

6.12.3.6 double Parameters::minpt

Minimal transverse momentum cut to apply on the outgoing lepton(s)

6.12.3.7 int Parameters::ntreat

Note

Is it correctly implemented?

6.12.3.8 int Parameters::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

6.12.3.9 int Parameters::p2mod

Note

Was named EMOD in ILPAIR

6.12.3.10 int Parameters::pair

The particle code of produced leptons, as defined by the PDG convention :

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

6.12.3.11 bool Parameters::symmetrise

List of Gnuplot objects which can be used to produce control plots all along the cross-section determination and events generation process

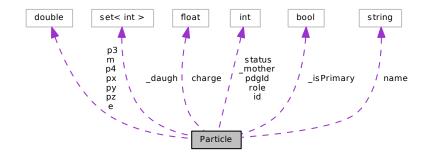
Note

Maximum number of these can be raised in the utils.h file, but pay attention to the memory load since these Gnuplot objects are still under development!

6.13 Particle Class Reference

Kinematics of one particle.

Collaboration diagram for Particle:



Public Member Functions

- Particle (int role_, int pdgld_=0)
 - Object constructor (providing the role of the particle in the process, and its Particle Data Group identifier)
- bool AddDaughter (Particle *part_)
 - Specify a decay product for this particle.
- void Dump ()
 - Dumps all the information on this particle.
- void E (double E_)
 - Sets the particle's energy.
- double E ()
 - Gets the particle's energy.
- double Eta ()
 - Pseudo-rapidity.

```
std::vector< int > GetDaughters ()
      Gets a vector containing all the daughters unique identifiers from this particle.
std::string GetLHEline (bool revert_=false)
int GetMother ()
      Gets the unique identifier to the mother particle from which this particle arises.
bool Hadronise (std::string algo_)
      Hadronises the particle using Pythia.
double M ()
      Gets the particle's mass.
bool M (double m_)
      Set the particle's mass in GeV/c^2.
double M2 ()
      Gets the particle's squared mass.
unsigned int NumDaughters ()
      Gets the number of daughter particles arising from this one.
bool operator< (const Particle &rhs)</li>
      Comparison operator to enable the sorting of particles in an event according to their unique identifier.
Particle & operator= (const Particle &)
      Copies all the relevant quantities from one Particle object to another.
bool P (double px_, double py_, double pz_)
      Sets the 3-momentum associated to the particle.

    bool P (double px_, double py_, double pz_, double E_)

      Sets the 4-momentum associated to the particle.
bool P (double p_[3], double E_)
      Sets the 4-momentum associated to the particle.

    bool P (double p_[4])

      Sets the 4-momentum associated to the particle.
double P ()
      Norm of the 3-momentum, in GeV/c.

    double * P3 ()

      Returns the particle's 3-momentum.

    double * P4 ()

      Returns the particle's 4-momentum.
  void PDF2PDG ()
double Phi ()
bool Primary ()
      Is this particle a primary particle?
double Pt ()
      Transverse momentum, in GeV/c.
double Rapidity ()
      Rapidity.
void SetMother (Particle *part_)
      Sets the mother particle (from which this particle arises)
bool Valid ()
      Is this particle a valid particle which can be used for kinematic computations?
```

Data Fields

float charge

The particle's electric charge (given as a float number, for the quarks and bound states)

int id

Unique identifier of the particle (in a **Event** object context)

std::string name

Particle's name in a human-readable format.

int pdgld

Particle Data Group integer identifier.

double px

Momentum along the x-axis in GeV/c.

double py

Momentum along the y-axis in GeV/c.

double pz

Momentum along the z-axis in GeV/c.

int role

Role in the considered process.

int status

Particle status.

Private Attributes

std::set< int > _daugh

List of daughter particles.

bool _isPrimary

Is the particle a primary particle?

int _mother

Mother particle.

■ double e

Energy, in GeV.

■ double m

Mass in GeV/c 2 .

- double **p3** [3]
- double **p4** [4]

6.13.1 Detailed Description

Kinematic information for one particle

6.13.2 Member Function Documentation

6.13.2.1 bool Particle::AddDaughter (Particle * part_)

Adds a "daughter" to this particle (one of its decay product(s))

Parameters

in part_	The Particle object in which this particle will desintegrate or convert	
----------	---	--

Returns

A boolean stating if the particle has been added to the daughters list or if it was already present before

6.13.2.2 void Particle::Dump ()

Dumps into the standard output stream all the available information on this particle

6.13.2.3 void Particle::E (double E_) [inline]

Parameters

in	E_	Energy, in GeV
----	----	----------------

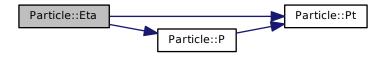
6.13.2.4 double Particle::Eta () [inline]

Computes and returns η , the pseudo-rapidity of the particle

Returns

The pseudo-rapidity of the particle

Here is the call graph for this function:



6.13.2.5 std::vector<int> Particle::GetDaughters ()

Returns

An integer vector containing all the daughters' unique identifier in the event

6.13.2.6 std::string Particle::GetLHEline (bool revert_ = false)

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

Parameters

in	revert_	Is the event symmetric ? If set to true, the third component of the mo-
		mentum is reverted.

Returns

The LHE line associated to the particle, and containing the particle's history (mother/daughters), its kinematics, and its status

6.13.2.7 int Particle::GetMother() [inline]

Returns

An integer representing the unique identifier to the mother of this particle in the event

6.13.2.8 bool Particle::Hadronise (std::string algo_)

Hadronises the particle with Pythia, and builds the shower (list of Particle objects) embedded in this object

Parameters

in	algo_	Algorithm in use to hadronise the particle

6.13.2.9 double Particle::M () [inline]

Gets the particle's mass in GeV/c 2 .

Returns

The particle's mass

6.13.2.10 bool Particle::P (double px_, double py_, double pz_) [inline]

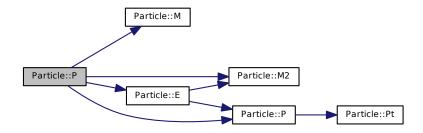
Parameters

in	px_	Momentum along the x -axis, in GeV/c
in	ру_	Momentum along the y -axis, in GeV/c
in	pz_	Momentum along the z -axis, in GeV/c

Returns

A boolean stating the validity of this particle (according to its 4-momentum norm)

Here is the call graph for this function:



6.13.2.11 bool Particle::P (double px_, double py_, double pz_, double E_) [inline]

Sets the 4-momentum associated to the particle, and computes its (invariant) mass.

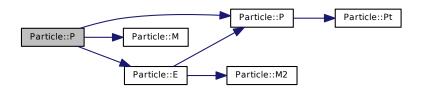
Parameters

in	px_	Momentum along the x -axis, in GeV/c
in	py_	Momentum along the y -axis, in GeV/c
in	pz_	Momentum along the z -axis, in GeV/c
in	E_	Energy, in GeV

Returns

A boolean stating the validity of the particle's kinematics

Here is the call graph for this function:



6.13.2.12 bool Particle::P (double $p_[3]$, double $E_$)

Parameters

in	p_	3-momentum
in	E_	Energy, in GeV

Returns

A boolean stating the validity of the particle's kinematics

6.13.2.13 bool Particle::P (double p_[4]) [inline]

Parameters

in p_ 4-momentum	
------------------	--

Returns

A boolean stating the validity of the particle's kinematics

Here is the call graph for this function:



6.13.2.14 double Particle::P() [inline]

Returns

The particle's 3-momentum norm as a double precision float

Here is the call graph for this function:



Returns

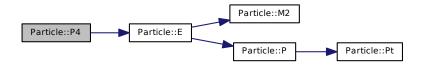
The particle's 3-momentum as a 3 components double array

Builds and returns the particle's 4-momentum as an array ordered as $(\mathbf{p}, E) = (p_x, p_y, p_z, E)$

Returns

The particle's 4-momentum as a 4 components double array

Here is the call graph for this function:



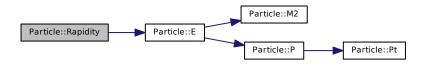
6.13.2.17 double Particle::Rapidity () [inline]

Computes and returns y, the rapidity of the particle

Returns

The rapidity of the particle

Here is the call graph for this function:



6.13.2.18 void Particle::SetMother (Particle * part_)

Sets the "mother" of this particle (particle from which this particle is issued)

Parameters

in	part	A Particle object containing all the information on the mother particle
	Purt_	1 71 dittele object containing an the information on the mother particle

6.13.3 Field Documentation

6.13.3.1 int Particle::pdgld

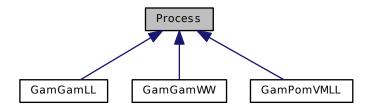
Unique identifier for a particle type. From [1]: The Monte Carlo particle numbering scheme [...] is intended to facilitate interfacing between event generators, detector simulators, and analysis packages used in particle physics.

6.13.3.2 int Particle::status

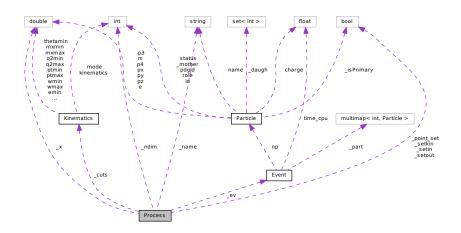
Codes 1-10 correspond to currently existing partons/particles, and larger codes contain partons/particles which no longer exist, or other kinds of event information

6.14 Process Class Reference

Inheritance diagram for Process:



Collaboration diagram for Process:



Public Member Functions

• virtual double ComputeWeight ()=0

Returns the weight for this point in the phase-space.

- virtual void DumpPoint ()
- virtual void FillKinematics (bool symmetrise_=false)

Fills the Event object with the particles' kinematics.

virtual Event * GetEvent ()

Returns the event content (list of particles with an assigned role)

- virtual std::string GetName ()
- virtual bool lsKinematicsDefined ()

Is the system's kinematics well defined?

- virtual unsigned int **ndim** () const
- virtual bool SetIncomingParticles (Particle ip1_, Particle ip2_)

Sets the momentum and PDG id for the incoming particles.

virtual void SetKinematics (Kinematics cuts_)

Sets the list of kinematic cuts to apply on the outgoing particles' final state.

virtual bool SetOutgoingParticles (int part_, int pdgld_)

Sets the PDG id for the outgoing particles.

virtual void SetPoint (const unsigned int ndim_, double x_[])

Sets the phase space point to compute.

virtual double x (const unsigned int idx_)

Protected Attributes

Kinematics _cuts

Set of cuts to apply on the final phase space.

Event * _ev

Event object containing all the information on the in- and outgoing particles.

- std::string __name
- unsigned int _ndim
- bool _point_set

Is the phase space point set ?

bool _setin

Are the event's incoming particles set ?

bool setkin

Is the full event's kinematic set?

bool _setout

Are the event's outgoing particles set ?

double * _x

6.14.1 Detailed Description

Class template to define any process to compute using this MC integrator/events generator

Author

Laurent Forthomme laurent.forthomme@uclouvain.be

Date

January 2014

6.14.2 Member Function Documentation

6.14.2.1 virtual void Process::FillKinematics (bool symmetrise_ = false) [inline], [virtual]

Fills the private Event object with all the Particle object contained in this event.

Parameters

in	symmetrise_	Do we have to symmetrise the event (randomise the production of the
		positively- and negatively-charged lepton) ?

Reimplemented in GamGamLL, GamGamWW, and GamPomVMLL.

```
6.14.2.2 virtual Event* Process::GetEvent ( ) [inline], [virtual]
```

Returns the complete list of Particle with their role in the process for the point considered in the phase space as an Event object.

Returns

The Event object containing all the generated Particle objects

```
6.14.2.3 virtual bool Process::IsKinematicsDefined ( ) [inline], [virtual]
```

Is the system's kinematics well defined and compatible with the process? This check is mandatory to perform the (_ndim)-dimensional point's cross-section computation.

Returns

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

```
6.14.2.4 virtual bool Process::SetIncomingParticles ( Particle ip1_, Particle ip2_ ) [inline], [virtual]
```

Specifies the incoming particles' kinematics as well as their properties using two Particle objects.

Parameters

in	ip1	Information on the first incoming particle
in	ip2_	Information on the second incoming particle

Returns

A boolean stating whether or not the incoming kinematics is properly set for this event

Reimplemented in GamGamLL, GamGamWW, and GamPomVMLL.

Here is the call graph for this function:



6.14.2.5 virtual void Process::SetKinematics (Kinematics cuts_) [inline], [virtual]

Parameters

in	cuts	The Cuts object containing the kinematic parameters
	_	, ,

Reimplemented in GamGamLL, GamGamWW, and GamPomVMLL.

6.14.2.6 virtual bool Process::SetOutgoingParticles (int part_, int pdgld_) [inline], [virtual]

Parameters

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

Returns

A boolean stating whether or not the outgoing kinematics is properly set for this event

Reimplemented in GamGamLL, GamGamWW, and GamPomVMLL.

Here is the call graph for this function:



6.14.2.7 virtual void Process::SetPoint (const unsigned int ndim_, double x_[]) [virtual]

Sets the phase space point to compute the weight associated to it.

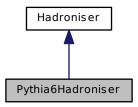
Parameters

in	ndim_	The number of dimensions of the point in the phase space
in	x_[]	The (ndim_)-dimensional point in the phase space on which the kinematics
		and the cross-section are computed

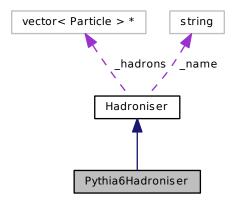
6.15 Pythia6Hadroniser Class Reference

Pythia6 hadronisation algorithm.

Inheritance diagram for Pythia6Hadroniser:



Collaboration diagram for Pythia6Hadroniser:



Public Member Functions

- std::vector< Particle > GetHadrons ()
- std::string GetName ()
- bool Hadronise (Particle *part_)

Main caller to hadronise a particle.

bool Hadronise (Event *ev_)

Hadronises a full event.

Protected Attributes

std::vector< Particle > * _hadrons

List of hadrons produced by this hadronisation process.

std::string __name

Name of the hadroniser.

Private Member Functions

bool PrepareHadronisation (Event *ev_)

Static Private Member Functions

- static void pyckbd ()
- static void pyexec ()
- static void pygive (const std::string &line_)
- static void pyjoin (int njoin_, int ijoin_[2])

Connect entries with colour flow information.

- static void pylist (int mlist_)
- static double pymass (int pdgid_)
- static std::string pyname (int pdgid_)
- static double pyp (int role_, int qty_)

6.15.1 Detailed Description

Full interface to the Pythia6 [3] algorithm. It can be used in a single particle decay mode as well as a full event hadronisation using the string model, as in Jetset.

6.15.2 Member Function Documentation

```
6.15.2.1 std::vector<Particle> Hadroniser::GetHadrons ( ) [inline], [inherited]
```

Gets the full list of hadrons (as Particle objects) produced by the hadronisation

Returns

A vector of Particle containing all the hadrons produced

```
6.15.2.2 bool Pythia6Hadroniser::Hadronise ( Event * ev_ ) [virtual]
```

Launches the hadroniser on the full event information

Parameters

in,out	ev_	The event to hadronise
--------	-----	------------------------

Returns

A boolean stating whether or not the hadronisation occured successfully

Reimplemented from Hadroniser.

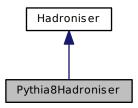
6.15.2.3 static void Pythia6Hadroniser::pyjoin (int njoin_, int ijoin_[2]) [inline], [static], [private]

Parameters

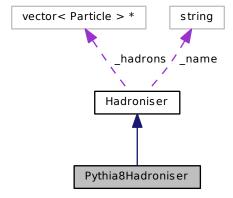
in	njoin_	Number of particles to join in the colour flow
in	ijoin_	List of particles unique identifier to join in the colour flow

6.16 Pythia8Hadroniser Class Reference

Inheritance diagram for Pythia8Hadroniser:



Collaboration diagram for Pythia8Hadroniser:



Public Member Functions

- std::vector< Particle > GetHadrons ()
- std::string GetName ()
- bool Hadronise (Event *ev_)

Hadronises a full event.

virtual bool Hadronise (Particle *part_)

Main caller to hadronise a particle.

Protected Attributes

std::vector< Particle > * _hadrons

List of hadrons produced by this hadronisation process.

std::string __name

Name of the hadroniser.

6.16.1 Member Function Documentation

6.16.1.1 std::vector<**Particle**> Hadroniser::GetHadrons () [inline], [inherited]

Gets the full list of hadrons (as Particle objects) produced by the hadronisation

Returns

A vector of Particle containing all the hadrons produced

6.16.1.2 bool Pythia8Hadroniser::Hadronise (**Event** * ev_) [virtual]

Launches the hadroniser on the full event information

Parameters

in,out	ev_	The event to hadronise
--------	-----	------------------------

Returns

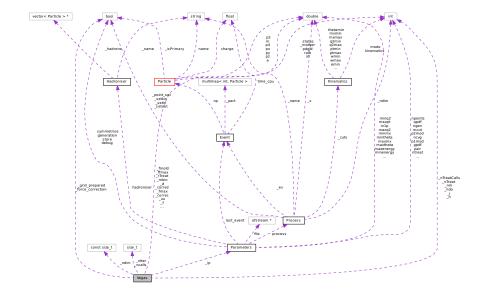
A boolean stating whether or not the hadronisation occured successfully

Reimplemented from Hadroniser.

6.17 Vegas Class Reference

Vegas Monte-Carlo integrator instance.

Collaboration diagram for Vegas:



```
Public Member Functions

    Vegas (const int dim_, double f_(double *, size_t, void *), Parameters *inParam_)

    ~Vegas ()

         Class destructor.
   void Generate ()
         Launches the generation of events.
   bool GenerateOneEvent ()
         Generates one single event according to the method defined in the Fortran 77 version of LPAIR.
   int Integrate (double *result_, double *abserr_)
Private Member Functions
   void DumpGrid ()
   double F (double *x_)
   double F (double *x_, Parameters *ip_)
```

Prepare the class for events generation.

bool StoreEvent (double *)

void SetGen ()

Stores the event in the output file.

- double Treat (double *x_, Parameters *ip_, bool storedbg_=false)
- double Treat (double *x_)
- double Treat (double *x_, bool storedbg_)

Private Attributes

```
double _corre2
double _correc

    double * _d [MAX_ND]

double * _di [MAX_ND]
double(* _f )(double *x_, size_t ndim_, void *params_)
double _ffmax
double * _fmax
double _fmax2
double _fmdiff
double _fmold
bool _force_correction
bool <u>grid_prepared</u>
Parameters * _ip
■ int __j
double _mbin
• int * _n
size_t _ncalls
     Fixed number of function calls to use.
const size_t _ndim
     The number of dimensions on which to integrate the function.
unsigned int _ndo
 size_t _nlter
     Number of points to generate in order to integrate the function.
 int * __nm
```

int _nTreat int _nTreatCalls double _rTreat

- double _weight
- double * _xi [MAX_ND]
- double * _xl

Lower bounds for the points to generate.

double * _xu

Upper bounds for the points to generate.

6.17.1 Detailed Description

Main occurence of the Monte-Carlo integrator[2] developed by G.P. Lepage in 1978

6.17.2 Constructor & Destructor Documentation

```
6.17.2.1 Vegas::Vegas ( const int dim_, double f_double *, size_t, void *, Parameters * inParam_ )
```

Constructs the class by booking the memory and structures for the Vegas integrator. This code is based on the Vegas Monte Carlo integration algorithm developed by P. Lepage, as documented in [2]

Parameters

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

6.17.3 Member Function Documentation

```
6.17.3.1 void Vegas::Generate ( )
```

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

```
6.17.3.2 bool Vegas::GenerateOneEvent ( )
```

Generates one event according to the grid parameters set in Vegas::SetGen

Returns

A boolean stating if the generation was successful (in term of the computed weight for the phase space point)

```
6.17.3.3 int Vegas::Integrate ( double * result_, double * abserr_ )
```

Vegas algorithm to perform the (_dim)-dimensional Monte Carlo integration of a given function as described in [2]

Author

```
Primary author : G.P. Lepage This C++ implementation : L. Forthomme
```

Date

September 1976 Reviewed in Apr 1978 FTN5 version 21 Aug 1984 This C++ implementation is from 12 Dec 2013

Parameters

out	result_	The cross section as integrated by Vegas for the given phase space restric-	
		tions	
out	abserr_	The error associated to the computed cross section	

Returns

0 if the integration was performed successfully

```
6.17.3.4 void Vegas::SetGen ( ) [private]
```

Sets all the generation mode variables and align them to the integration grid set while computing the cross-section

```
6.17.3.5 bool Vegas::StoreEvent ( double * ) [private]
```

Stores the event characterized by its *ndim-dimensional point in the phase space to the output file* x The _ndim-dimensional point in the phase space defining the unique event to store

Returns

A boolean stating whether or not the event could be saved

64 REFERENCES

References

- [1] J. Beringer et al. Review of Particle Physics (RPP). Phys.Rev., D86:010001, 2012. 53
- [2] G Peter Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192 203, 1978. 1, 62
- [3] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. PYTHIA 6.4 Physics and Manual. *JHEP*, 0605:026, 2006. 58
- [4] J.A.M. Vermaseren. Two-photon processes at very high energies. *Nuclear Physics B*, 229(2):347 371, 1983. 1, 12, 15, 16

Index

_alpha1	NumParticles, 9
GamPomVMLL, 30	Store, 9
_amxb0	·
GamPomVMLL, 30	FillKinematics
_anexp	GamGamLL, 18
GamPomVMLL, 30	GamGamWW, 23
_b0	GamPomVMLL, 27
GamPomVMLL, 30	Process, 55
_epsilm	
·	GEPhot
GamPomVMLL, 30	GamPomVMLL, 28
_epsilw	GamGamLL, 9
GamPomVMLL, 30	ComputeCMenergy, 18
_wmax	ComputeMX, 18
GamPomVMLL, 30	ComputeWeight, 18
AddDaughter	FillKinematics, 18
Particle, 48	GamGamLL, 17
AddParticle	GamGamLL, 17
Event, 5	GetEvent, 18
AnalyzePhaseSpace	GetT1, <mark>18</mark>
MCGen, 41	GetT1extrema, 19
	GetT2, 19
ComputeCMenergy	GetT2extrema, 19
GamGamLL, 18	IsKinematicsDefined, 19
ComputeMX	Orient, 19
•	PeriPP, 19
GamGamLL, 18	Pickin, 19
ComputeWeight	
GamGamLL, 18	PrepareHadronisation, 20
ComputeXsection	SetIncomingParticles, 20
MCGen, 41	SetKinematics, 20
	SetOutgoingParticles, 20
debug	SetPoint, 20
Parameters, 45	GamGamWW, 21
DecVM	FillKinematics, 23
GamPomVMLL, 27	GetEvent, 23
Dump	IsKinematicsDefined, 23
Event, 5	SetIncomingParticles, 23
Particle, 48	SetKinematics, 23
r division, no	SetOutgoingParticles, 24
Е	SetPoint, 24
Particle, 49	GamPomVMLL, 24
Eta	_alpha1, 30
Particle, 49	_amxb0, 30
Event, 3	_anexp, 30
AddParticle, 5	_b0, 30
Dump, 5	_epsilm, 30
GetByld, 5	_epsilw, 30
GetBylds, 5	_wmax, 30
GetByRole, 6	DecVM, 27
GetDaughters, 6	FillKinematics, 27
GetLHERecord, 6	GEPhot, 28
GetMother, 6	GetEvent, 28
GetOneByRole, 8	IsKinematicsDefined, 28
GetParticles, 8	itypvm, 30
GetRoles, 8	SetIncomingParticles, 28
GetNoies, 6 GetStableParticles, 9	SetKinematics, 29
GetStablet articles, 3	Jenninghes, 29

66 INDEX

SetOutgoingParticles, 29	Pythia6Hadroniser, 58
SetPoint, 29	Pythia8Hadroniser, 60
VXMass, 29	Hadroniser, 31
Generate	GetHadrons, 32
Vegas, 62	Hadronise, 32
GenerateOneEvent	Herwig6Hadroniser, 35
MCGen, 42	GetHadrons, 36
Vegas, 62	Hadronise, 36
GetByld	
Event, 5	Integrate
GetBylds	Vegas, 62
Event, 5	IsKinematicsDefined
GetByRole	GamGamLL, 19
Event, 6	GamGamWW, 23
GetDaughters	GamPomVMLL, 28
Event, 6	Process, 55
Particle, 49	itypvm
GetEvent	GamPomVMLL, 30
GamGamLL, 18	
GamGamWW, 23	Jetset7Hadroniser, 36
GamPomVMLL, 28	GetHadrons, 38
Process, 55	Hadronise, 38
GetHadrons	lujoin, 38
Hadroniser, 32	K:
Herwig6Hadroniser, 36	Kinematics, 38
Jetset7Hadroniser, 38	kinematics, 39
Pythia6Hadroniser, 58	kinematics
Pythia8Hadroniser, 60	Kinematics, 39
GetLHERecord	LaunchGeneration
Event, 6	MCGen, 42
GetLHEline	lujoin
Particle, 49	Jetset7Hadroniser, 38
GetMother	Jetset/Hadromser, 36
Event, 6	M
Particle, 49	Particle, 50
GetOneByRole	MCGen, 40
Event, 8	AnalyzePhaseSpace, 41
GetParticles	ComputeXsection, 41
Event, 8	GenerateOneEvent, 42
GetRoles	LaunchGeneration, 42
Event, 8	MCGen, 41
GetStableParticles	MCGen, 41
Event, 9	maxmx
GetT1	Parameters, 45
GamGamLL, 18	maxpt
Get T1extrema	Parameters, 45
GamGamLL, 19	mcut
Get T2	Parameters, 45
GamGamLL, 19	minmx
GetT2extrema	Parameters, 45
GamGamLL, 19	minpt
Ja Ja,	Parameters, 45
HEPEUP, 33	i didilicters, 73
HEPRUP, 34	ntreat
Hadronise	Parameters, 45
Hadroniser, 32	NumParticles
Herwig6Hadroniser, 36	Event, 9
Jetset7Hadroniser, 38	
Particle, 49	Orient

INDEX 67

GamGamLL, 19 Particle, 50, 51 plmod Parameters, 45 P2 Parameters, 45 P3 Particle, 52 P4 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Particle, 52 Particle, 52 Particle, 52 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 46 Parameters, 42 debug, 45 maxmx, 45 maxpt, 45 minmx, 45 minmx, 45 minmx, 45 minmt, 45 ninpt, 45 ninmt, 45 pilmod, 45 pilmod		
Particle, 50, 51 plmod Parameters, 45 p2mod Parameters, 45 P3 Particle, 52 P4 Particle, 52 Particle, 52 Parmeters, 45 Parmeters, 45 Parmeters, 45 Particle, 52 Particle, 52 Parmeters, 45 Parmeters, 45 Parmeters, 45 Parameters, 45 Parameters, 45 Parmeters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 46 Parameters, 47 Parameters, 48 Parameters, 49 Parameters, 49 Parmeters, 49 Parmeters, 44 SetEtaRange Parameters, 44 SetEtaRange Parameters, 44 SetGen Vegas, 63 SetIncomingParticles GamGamLL, 20 GamGamWW, 23 GamPomVMLL, 28 Process, 55 SetNother, 49 GetDaughter, 48 Dump, 48 E, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 Particle, 53 Particle, 53 Particle, 53 Particle, 53 Particle, 53 Particle, 53 PeriPP GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Vegas, 60 Pegas, 60 Vegas, 60	GamGamLL, 19	SetKinematics, 56
Particle, 50, 51 plmod Parameters, 45 p2mod Parameters, 45 P3 Patricle, 52 P4 Particle, 52 Parmeters, 45 Parmeters, 45 Parmeters, 45 Particle, 52 Parmeters, 45 Parmeters, 45 Parmeters, 45 Parmeters, 45 Parmeters, 45 Parmeters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 46 Parameters, 47 Parmeters, 48 Parmeters, 49 Parmeters, 49 Parmeters, 45 Parmeters, 45 Parmeters, 45 Parmeters, 46 Parameters, 46 Parameters, 46 Parmeters, 47 Particle, 52 ReadConfigFile Parameters, 44 Parmeters, 49 Particle, 53 Parameters, 44 Parameters, 44 Parmeters, 45 Parameters, 46 Para	P	
p1mod Parameters, 45 P2mod Parameters, 45 P3 Particle, 52 Particle, 52 Particle, 52 Parameters, 45 Particle, 52 Particle, 52 Parameters, 45 Particle, 52 Particle, 52 Parameters, 45 Parameters, 42 debug, 45 maxmx, 45 maxmx, 45 maxmx, 45 maxpt, 45 minpt, 45 ntreat, 45 plmod, 45 plmod, 45 plmod, 45 plmod, 45 pair, 45 ReadConfigFile, 44 SetGen vegas, 63 SetIncomingParticles GamGamLl, 20 GamGamWW, 23 GamPomVMLL, 28 Process, 55 SetKinematics SetKinematics SetKinematics SetWother Particle, 46 AddDaughter, 48 Dump, 48 E, 49 SetDaughters, 49 GetDaughters, 49 GetDaughters, 49 GetLHEline, 49 GetLHEline, 49 GetLHEline, 49 GetLHeline, 49 GetLHeline, 49 GetLHeline, 49 GetMother, 49 Hadronise, 49 Process, 56 SetPoint GamGamLl, 20 Particle, 53 SetOutgoingParticles GamGamLl, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLl, 20 GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamUL, 20 Process, 56 SetMother, 53 status, 53 pdgld Particle, 53 PeriPP GamGamLl, 19 Process, 56 SetMother, 53 status, 53 pdgld Particle, 53 Particle, 53 Particle, 53 Particle, 53 SetMother, 53 status, 53 pdgld Particle, 53 Particle, 53 Particle, 53 Particle, 53 SetMinematics, 55 GetEvent, 55 Iskinematics, 55 GetEvent, 55 Iskinematics, 55 GetEvent, 55 Iskinematics, 55 SetMomputalle, 29 Vegas, 60 VXMass GamPomVMLL, 29 Vegas, 60		
Parameters, 45 P2mod Parameters, 45 P3 Particle, 52 P4 Particle, 52 Pair Parameters, 45 Parameters, 42 debug, 45 maxmx, 45 maxmx, 45 maxmx, 45 minnt, 45 minnt, 45 minnt, 45 minnt, 45 minnt, 45 pair, 45 pair, 45 Parameters, 49 Parameters, 44 SetEaRange Parameters, 44 SetGen Vegas, 63 Parameters, 44 SetEtaRange Parameters, 44 SetEtaRange Parameters, 44 SetGen Vegas, 63 SetIncomingParticles GamGamLL, 20 GamGamWW, 23 GamPomVMLL, 28 Process, 55 SetKinematics GamGamLL, 20 GamGamMW, 23 GamPomVMLL, 28 Process, 56 SetMother Particle, 46 GamGamL, 20 GamGamWW, 23 GamPomVMLL, 29 Process, 56 SetMother Particle, 53 SetOutgoingParticles GamGamLL, 20 GamGamWW, 23 GamGamL, 20 GamGamWW, 23 GamGamWW, 23 GamGamWW, 23 GamGamL, 20 GamGamWW, 23 GamGamL, 20 GamGamL, 20 GamGamL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetMother Particle, 53 SetOutgoingParticles GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 Forcess, 55 SetMother, 53 Status		
p2mod	·	
Parameters, 45 Parameters, 45 Particle, 52 P4 Particle, 52 P4 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 42 Parameters, 42 ReadConfigFile Parameters, 45 Parameters, 45 Parameters, 45 ReadConfigFile Parameters, 44 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 46 Parameters, 45 Parameters, 45 Parameters, 46 Parameters, 44 SetGen Vegas, 63 Parameters, 44 SetGen Vegas, 63 Parameters, 44 SetLaRange Parameters, 45 SetLincomingParticles GamGamWW, 23 GamPomVMLL, 28 Proces, 55 SetKinematics SetMominatics SetMomantics GamGamWW, 23 GamGamWW, 24 GamGamLt, 29 Process, 56 SetPoint GamGamWW, 24 GamGamGamWW, 24 GamGamWW, 25 GamGamWW, 24 GamGamWW, 25 GamGamWW, 26 GamGamW		· · · · · · · · · · · · · · · · · · ·
P3 P4 Particle, 52 P4 Particle, 52 Pir Parmeters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 46 Particle, 52 debug, 45 maxmx, 45 maxmx, 45 minpt, 45 minpt, 45 minpt, 45 ntreat, 45 plmod, 45 ppir, 45 pair, 45 ReadConfigFile Parameters, 44 Parameters, 44 minpt, 45 minpt, 45 ntreat, 45 plmod, 45 pair, 45 ReadConfigFile, 44 SetEtaRange Parameters, 44 SetEtaRange Negas, 63 SetIncomingParticles GamGamLL, 20 GamGamWW, 23 GamPomVMLL, 28 Process, 55 SetKinematics SetKinematics SetKinematics SetKinematics SetKinematics SetKinematics SetKinematics SetMother Particle, 46 Particle, 46 Particle, 46 Particle, 49 GetDaughters, 49 GetDaughters, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother, 53 status, 53 Particle, 53 SetPoint GamGamLL, 19 PrepareHadronisation GamGamLL, 19 PrepareHadronisation GamGamLL, 19 PrepareHadronisation GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Parameters, 46 Parameters, 49 Parameters, 40 P	•	
Particle, 52 Particle, 52 Particle, 52 Particle, 52 Particle, 52 Parameters, 45 Parameters, 45 Parameters, 45 Parameters, 42 debug, 45 maxmx, 45 maxmx, 45 mcut, 45 mcut, 45 minnt, 45 minnt, 45 minnt, 45 ntreat, 45 pland, 55 pland, 51 pland, 51 pland, 52 pland, 53 pland, 52 pland, 53 pland, 52 pland, 53 pland, 53 pland, 52 pland, 53 pl		
P4 Particle, 52 Paricle, 52 Parameters, 45 Parameters, 42		
Particle, 52 pair Parameters, 45 Parameters, 42 debug, 45 maxmx, 45 maxpt, 45 minnx, 45 minnx, 45 minpt, 45 ntreat, 45 pair, 46 particle, 49 particle, 53 pair, 45 pair,	•	· · · · · · · · · · · · · · · · · · ·
pair Parameters, 45 Parameters, 42 debug, 45 maxmx, 45 maxmx, 45 maxmx, 45 minmx, 45		
Parameters, 45 Rapidity Parameters, 42 Particle, 52 debug, 45 ReadConfigFile maxpt, 45 Parameters, 44 maxpt, 45 SetEtaRange minmx, 45 Parameters, 44 minpt, 45 SetGen ntreat, 45 Vegas, 63 plmod, 45 SetIncomingParticles pair, 45 GamGamLL, 20 pair, 45 GamGamWW, 23 ReadConfigFile, 44 GamPomVMLL, 28 SetEtaRange, 44 Process, 55 StoreConfigFile, 44 SetKinematics Symmetrise, 46 GamGamLL, 20 Particle, 46 GamGamLL, 20 Particle, 46 GamGamLL, 20 Particle, 46 GamGamLL, 20 Dump, 48 Process, 56 E, 49 SetMother Eta, 49 SetOutgoingParticles GetDaughters, 49 GamGamLL, 20 GetMother, 49 GamGamWW, 24 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamWW, 24 GamPomVMLL, 29		Hadronise, 60
Parameters, 42 Particle, 52 debug, 45 ReadConfigFile maxmx, 45 Parameters, 44 maxpt, 45 Parameters, 44 mcut, 45 SetEtaRange minmx, 45 Parameters, 44 minpt, 45 SetGen vegas, 63 SetIncomingParticles pain, 45 GamGamLL, 20 pair, 45 GamGamWW, 23 ReadConfigFile, 44 GamGamWW, 23 SetEtaRange, 44 SetEitaRange, 44 StoreConfigFile, 44 SetMommund symmetrise, 46 GamGamLL, 20 Particle, 46 GamGamLL, 20 AddDaughter, 48 Process, 55 Dump, 48 Process, 56 E, 49 SetMother Eta, 49 SetMother GetDaughters, 49 SetOutgoingParticles GetHelline, 49 GamGamLL, 20 GetMother, 49 GamGamWW, 24 Hadronise, 49 GamGamMW, 24 M, 50 Process, 56 P, 50, 51 SetPoint GamGamLL, 20 GamGamBMW, 24	•	5
debug, 45 maxmx, 45 maxmx, 45 maxmx, 45 maxmt, 45 muct, 45 minmx, 45 min		• •
maxmx, 45 maxpt, 45 mcut, 45 mcut, 45 miny, 45 miny, 45 minpt, 45	•	
maxpt, 45 mcut, 45 mcut, 45 minmx, 45 minmx, 45 minpt, 45 minpt, 45 plmod, 45 plmod, 45 pair, 45 SetGen vegas, 63 plmod, 45 pair, 45 ReadConfigFile, 44 SetEtaRange, 46 SetKinematics GamGamLL, 20 GamGamWW, 23 GamPomVMLL, 28 Process, 55 SetKinematics GamGamLL, 20 GamGamWW, 23 GamGamWW, 24 SetKinematics GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetMother Particle, 53 SetOutgoingParticles GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetBother, 53 Status Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 Vegas, 60 Vegas, 60		<u>~</u>
mcut, 45 SetEtaRange minmx, 45 Parameters, 44 minpt, 45 SetGen ntreat, 45 Vegas, 63 plmod, 45 SetIncomingParticles pair, 45 GamGamUL, 20 pair, 45 GamPomVMLL, 28 ReadConfigFile, 44 SetEtaRange, 44 StoreConfigFile, 44 SetKinematics symmetrise, 46 GamGamLL, 20 Particle, 46 GamGamWW, 23 AddDaughter, 48 GamPomVMLL, 29 Dump, 48 Process, 56 E, 49 SetMother Eta, 49 Particle, 53 GetDaughters, 49 SetOutgoingParticles GetMother, 49 GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamWW, 29 Process, 56 SetPoint GamGamWW, 24 GamGamWW, 24 GamPomVMLL, 29 P4, 52 GamGamWW, 24 pdgld, 53 Particle, 53 Rapidity, 52 SetMother, 53 status Farameters, 44		Parameters, 44
minmx, 45 minpt, 45 minpt, 45 ntreat, 45 plmod, 45 plmod, 45 pair, 45 ReadConfigFile, 44 SettEaRange, 44 SettEtaRange, 44 SetEtaRange, 44 SetWinematics SetWinematics SetWinematics SetKinematics SetM		C . E. D
minpt, 45 ntreat, 45 ntreat, 45 p1mod, 45 p2mod, 45 pair, 45 ReadConfigFile, 44 SetEtaRange, 44 StoreConfigFile, 44 Summetrise, 46 Particle, 46 Particle, 46 Particle, 49 GetDaughters, 49 GetLHEline, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother, 53 status, 53 pdgld Particle, 53 PeriPP GamGamLL, 19 PrepareHadronisation GamGamLL, 20 GamGamLL, 20 GamGamWW, 24 SetMother Particle, 53 SetOutgoingParticles GamGamLL, 20 GamGamWW, 23 SetMother Particle, 53 SetOutgoingParticles GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 Status Particle, 53 Status Particle, 53 Status Particle, 53 Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 Vegas, 60		<u> </u>
ntreat, 45 p1mod, 45 p2mod, 45 p2mod, 45 pair, 45 ReadConfigFile, 44 SetEtaRange, 44 SetEtaRange, 44 SourceonfigFile, 45 SourceonfigFile, 46 SourceonfigFile, 48 SourceonfigFile, 49 SourceonfigFile SourceonfigFile Parameters, 44 SourceonfigFile Parameters, 44 StoreEvent SourceonfigFile Parameters, 44 StoreEvent SourceonfigFile Parameters, 46 Process, 53 FillKinematics, 55 SourceonfigFile Parameters, 46 S		•
p1mod, 45 p2mod, 45 pair, 45 ReadConfigFile, 44 SetEtaRange, 44 SetEtaRange, 44 Sommetrise, 46 Particle, 46 AddDaughter, 48 Dump, 48 Eta, 49 GetDaughters, 49 GetLHEline, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother, 53 SetMother, 53 SetMother, 53 SetMother Particle, 53 Rapidity, 52 SetMother, 53 SetMother Particle, 53 SetPoint Particle, 53 SetPoint GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint Particle, 53 SetMother, 29 Process, 56 SetPoint Particle, 53 SetPoint SamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint Particle, 53 SetMother, 53 Status, 53 Particle, 53 SetMother, 53 Status, 53 Particle, 53 SetPoint SamGamLL, 20 SamGamWW, 24 SamPomVMLL, 29 Process, 56 SetPoint SamGamUW, 24 SamPomVMLL, 29 SetMother, 53 Status, 53 Status, 53 SetPoint SamGamLL, 20 SamGamWW, 24 SamPomVMLL, 29 Process, 56 Status Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 Process, 55 SetEvent, 55 SetMilkinematics, 55 SetEvent, 55 SetMother, 53 Store Status Status StoreEvent Vegas, 63 Symmetrise StoreEvent Vegas, 63 Symmetrise SamPomVMLL, 29 StoreSomPomVMLL, 29 StoreSomPomVML, 29 StoreSomPomVML, 29 StoreSomPomVML, 29 StoreSomPomVML, 20 StoreSomPomVML, 20 StoreSomPomVML, 20 StoreSomPomVML, 20 StoreSomPomVML, 20 Sto	•	
p2mod, 45 pair, 45 pair, 45 ReadConfigFile, 44 SetEtaRange, 44 StoreConfigFile, 44 StoreConfigFile, 44 Symmetrise, 46 Particle, 46 AddDaughter, 48 Dump, 48 E, 49 Eta, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 P3, 52 pdgld, 53 Rapidity, 52 SetMother, 53 status, 53 pdgld Particle, 53 PeriPP GamGamLL, 19 Process, 56 Store Event, 9 StoreConfigFile AddDaughter, 49 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint Status Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 Process, 55 SetEvent, 9 SetEvent, 9 SetEvent, 9 SetEvent, 9 SetEvent, 9 SetEvent, 9 Se		
pair, 45 ReadConfigFile, 44 SetEtaRange, 44 StoreConfigFile, 44 symmetrise, 46 Particle, 46 AddDaughter, 48 Dump, 48 Eta, 49 Eta, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother, 53 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetMother, 53 Status Status, 53 PeriPP StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise GamGamLL, 20 Process, 55 IsKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 VXMass GamPomVMLL, 29 Vegas, 60	•	<u> </u>
ReadConfigFile, 44 SetEtaRange, 44 SetEtaRange, 44 Symmetrise, 46 Symmetrise, 46 Particle, 46 AddDaughter, 48 Dump, 48 Eta, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother, 53 SetMother, 53 SetBuggld Particle, 53 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetMother, 53 Status Store Farticle, 53 PeriPP StoreConfigFile Parameters, 44 StoreEvent GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 VXMass GamPomVMLL, 29 Vegas, 60	•	
SetEtaRange, 44 Process, 55 StoreConfigFile, 44 SetKinematics symmetrise, 46 GamGamLL, 20 Particle, 46 GamGamWW, 23 AddDaughter, 48 GamPomVMLL, 29 Dump, 48 Process, 56 E, 49 SetMother Eta, 49 Particle, 53 GetDaughters, 49 SetOutgoingParticles GetLHEline, 49 GamGamWW, 24 GetMother, 49 GamGamWW, 24 Hadronise, 49 GamGamWW, 24 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamLL, 20 P4, 52 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Process, 56 Status status, 53 Particle, 53 pdgld Store Particle, 53 Status periPP StoreConfigFile GamGamLL, 19 Parameters, 44 Pickin StoreEvent GamGamLL, 20 Parameters, 46 Process, 53 FillKinematics, 55 GamPomVMLL, 29	•	
StoreConfigFile, 44 symmetrise, 46 Particle, 46 AddDaughter, 48 Dump, 48 E, 49 GetLa, 49 GetLa, 49 GetLaughter, 49 GetLHEline, 49 GetMother, 49 GetMother, 49 Hadronise, 49 M, 50 Process, 56 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother, 53 SetMother, 53 SetMother, 53 SetPoint GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamLL, 20 GamGamWW, 24 Process, 56 SetPoint GamGamLL, 20 GamGamWW, 24 SetPoint GamGamWW, 24 SetPoint GamGamLL, 20 GamGamWW, 24 SetPoint GamGamWW, 24 SetPoint GamGamLL, 20 SetPoint GamGamWW, 24 SetPoint GamGamL, 50 SetPoint GamGamUL, 20 SetPoint StoreConfigFile Parameters, 46 StoreEvent Vegas, 63 Symmetrise GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 VXMass GamPomVMLL, 29 Vegas, 60		
symmetrise, 46 GamGamLL, 20 Particle, 46 GamGamWW, 23 AddDaughter, 48 GamPomVMLL, 29 Dump, 48 Process, 56 E, 49 SetMother Eta, 49 Particle, 53 GetDaughters, 49 SetOutgoingParticles GetHEline, 49 GamGamLL, 20 GetMother, 49 GamGamWW, 24 Hadronise, 49 GamPomVMLL, 29 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamLL, 20 P4, 52 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Rapidity, 52 Process, 56 SetMother, 53 status status, 53 Particle, 53 PeriPP StoreConfigFile GamGamLL, 19 Parameters, 44 Pickin StoreEvent GamGamLL, 19 Vegas, 63 Process, 53 Parameters, 46 Process, 53 FillKinematics, 55 GetEvent, 55 GamPomVMLL, 29 Vegas, 60 Vegas, 60		•
Particle, 46 GamGamWW, 23 AddDaughter, 48 GamPomVMLL, 29 Dump, 48 Process, 56 E, 49 SetMother Eta, 49 SetOutgoingParticles GetDaughters, 49 GamGamL, 20 GetMother, 49 GamGamWW, 24 Hadronise, 49 GamPomVMLL, 29 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamL, 20 GamGamWW, 24 GamPomVMLL, 29 Pdgld, 53 GamGamWW, 24 Rapidity, 52 Process, 56 SetMother, 53 status status, 53 Particle, 53 PeriPP Store GamGamLL, 19 Parameters, 44 Pickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation symmetrise GamGamLL, 20 Parameters, 46 Process, 53 FillKinematics, 55 GetEvent, 55 GamPomVMLL, 29 Vegas, 60 Vegas, 60	<u> </u>	
AddDaughter, 48 Dump, 48 Pump, 48 Process, 56 E, 49 Eta, 49 Eta, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 Process, 56 P, 50, 51 P3, 52 P4, 52 pdgld, 53 Rapidity, 52 SetMother GamGamUL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamWW, 24 GamGamWW, 24 GamGamUL, 20 GamGamWW, 24 GamGamUL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamUL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetWother, 53 Status Status Status Status Status Status Status Status Store Particle, 53 PeriPP GamGamLL, 19 Pickin GamGamLL, 19 PrepareHadronisation GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 VXMass GamPomVMLL, 29 Vegas, 60	•	
Dump, 48 Process, 56 E, 49 SetMother Eta, 49 Particle, 53 GetDaughters, 49 SetOutgoingParticles GetMother, 49 GamGamLL, 20 Hadronise, 49 GamPomVMLL, 29 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamLL, 20 P4, 52 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Rapidity, 52 Process, 56 SetMother, 53 status status, 53 Particle, 53 PeriPP Store GamGamLL, 19 Parameters, 44 Pickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation symmetrise GamGamLL, 20 Parameters, 46 Process, 53 VXMass FillKinematics, 55 GamPomVMLL, 29 Vegas, 60 Vegas, 60		
E, 49 SetMother Eta, 49 Particle, 53 GetDaughters, 49 SetOutgoingParticles GetLHEline, 49 GamGamLL, 20 GetMother, 49 GamGamWW, 24 Hadronise, 49 Process, 56 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamLL, 20 P4, 52 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Rapidity, 52 Process, 56 SetMother, 53 status status status status Store Particle, 53 Particle, 53 PeriPP StoreConfigFile GamGamLL, 19 Parameters, 44 Process, 53 Symmetrise GamGamLL, 20 Parameters, 46 Process, 53 VXMass FillKinematics, 55 GamPomVMLL, 29 IsKinematicsDefined, 55 Vegas, 60	_	
Eta, 49 GetDaughters, 49 GetLHEline, 49 GetMother, 49 Hadronise, 49 M, 50 Process, 56 P, 50, 51 Pdgld, 53 SetPoint Particle, 53 SetMother, 53 SetBoint SetPoint Set	•	
GetDaughters, 49 GetLHEline, 49 GetMother, 49 GetMother, 49 Hadronise, 49 M, 50 Process, 56 P, 50, 51 P3, 52 P4, 52 Pdgld, 53 Rapidity, 52 SetMother, 53 Status, 53 PeriPP GamGamLL, 19 Prickin GamGamLL, 19 Process, 53 FillKinematics, 55 GetEvent, 55 IskinematicsDefined, 55 SetOutgoingParticles GamGamLL, 20 GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 SetPoint GamGamLL, 20 GamGamLL, 20 GamGamLL, 20 GamGamWW, 24 GamPomVMLL, 29 Process, 56 Status Status Status Status Status Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 Process, 53 FillKinematics, 55 GamPomVMLL, 29 Vegas, 60		
GetLHEline, 49 GamGamLL, 20 GetMother, 49 GamGamWW, 24 Hadronise, 49 GamPomVMLL, 29 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamLL, 20 P4, 52 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Rapidity, 52 Process, 56 SetMother, 53 status status, 53 Particle, 53 PeriPP Store GamGamLL, 19 Parameters, 44 Pickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation symmetrise GamGamLL, 20 Parameters, 46 Process, 53 VXMass FillKinematics, 55 GamPomVMLL, 29 IsKinematicsDefined, 55 Vegas, 60		
GetMother, 49 Hadronise, 49 M, 50 Process, 56 P, 50, 51 P3, 52 GamGamWW, 24 Pdgld, 53 Rapidity, 52 SetMother, 53 Status, 53 Pdgld Particle, 53 PeriPP GamGamLL, 19 Pickin GamGamLL, 19 Process, 56 Status Status GamGamLL, 19 Process, 56 Store Parameters, 44 StoreEvent GamGamLL, 19 Process, 53 FillKinematics, 55 GetEvent, 55 IskinematicsDefined, 55 GamPomVMLL, 29 Frocess, 56 Status Status Farticle, 53 Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
Hadronise, 49 GamPomVMLL, 29 M, 50 Process, 56 P, 50, 51 SetPoint P3, 52 GamGamLL, 20 P4, 52 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Rapidity, 52 Process, 56 SetMother, 53 status status, 53 Particle, 53 PeriPP Store GamGamLL, 19 Parameters, 44 Pickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation symmetrise GamGamLL, 20 Parameters, 46 Process, 53 VXMass GetEvent, 55 GamPomVMLL, 29 IskinematicsDefined, 55 Vegas, 60	•	
M, 50 P, 50, 51 SetPoint P3, 52 P4, 52 P4, 52 Pdgld, 53 Rapidity, 52 SetMother, 53 Status, 53 Particle, 53 PeriPP Parameters, 44 Pickin GamGamLL, 19 Process, 56 Store Fardial, 19 Process, 56 Store FerepareHadronisation GamGamLL, 19 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Perpore SetPoint GamGamLL, 19 Process, 56 SetPoint GamGamLL, 29 Frocess, 56 SetPoint GamGamVW, 24 GamGamVW, 24 GamPomVMLL, 29 Frocess, 56 SetPoint GamPomVMLL, 29 Vegas, 63 Store Forcess, 53 VXMass GamPomVMLL, 29 Vegas, 60		
P, 50, 51 P3, 52 P4, 52 P4, 52 Pdgld, 53 Rapidity, 52 SetMother, 53 Status, 53 Pdgld Particle, 53 PeriPP GamGamLL, 19 Pickin GamGamLL, 19 Pickin GamGamLL, 19 Process, 53 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 SetPoint GamGamLL, 20 GamGamLL, 20 SetPoint GamGamLL, 20 GamGamLL, 20 Focess, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 SetPoint GamGamLL, 20 GamGamVW, 24 GamPomVMLL, 29 Vegas, 66 GamPomVMLL, 29 Vegas, 60		
P3, 52 P4, 52 P4, 52 GamGamWW, 24 pdgld, 53 Rapidity, 52 SetMother, 53 status, 53 Particle, 53 PeriPP Process, 56 GamGamLL, 19 Pickin GamGamLL, 19 Process, 53 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 GamGamWW, 24 GamGamWW, 24 GamGamWW, 24 GamGamWM, 29 Process, 56 Status Store Particle, 53 Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 VXMass GamGamLL, 20 Vegas, 60		•
P4, 52 pdgld, 53 GamGamWW, 24 pdgld, 53 GamPomVMLL, 29 Rapidity, 52 SetMother, 53 status, 53 pdgld		
pdgld, 53 Rapidity, 52 Rapidity, 52 SetMother, 53 status, 53 Particle, 53 Particle, 53 PeriPP PeriPP GamGamLL, 19 Pickin GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 RamPomVMLL, 29 Process, 56 Status Particle, 53 Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		•
Rapidity, 52 SetMother, 53 SetMother, 53 Status, 53 pdgld Particle, 53 PeriPP PeriPP StoreConfigFile Parameters, 44 Pickin GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Process, 53 FillKinematicsDefined, 55 Process, 53 Setatus Store Event, 9 StoreConfigFile Parameters, 44 Vegas, 63 StoreEvent Vegas, 63 Symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
SetMother, 53 status, 53 pdgld Particle, 53 PeriPP PeriPP GamGamLL, 19 Pickin GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Store Event, 9 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
status, 53 pdgld Particle, 53 Store Particle, 53 PeriPP StoreConfigFile GamGamLL, 19 Pickin GamGamLL, 19 Vegas, 63 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Particle, 53 Perticle, 53 Perticle, 53 Perent, 9 StoreConfigFile Parameters, 44 Parameters, 44 Process, 63 Symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
pdgld Store Particle, 53 Event, 9 PeriPP StoreConfigFile GamGamLL, 19 Prickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation GamGamLL, 20 Parameters, 46 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Store Parameters, 44 StoreEvent Vegas, 63 Vegas, 63 Vegas, 63 Vogas, 63 Vogas, 63 Vegas, 63 Vogas, 63 Vegas, 60		
Particle, 53 PeriPP StoreConfigFile GamGamLL, 19 Pickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 PrepareHadronisation Symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
PeriPP GamGamLL, 19 Pickin GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 StoreConfigFile Parameters, 44 StoreEvent Vegas, 63 Symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60	• •	
GamGamLL, 19 Pickin StoreEvent GamGamLL, 19 Vegas, 63 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Parameters, 44 Vegas, 63 Symmetrise Parameters, 46 Values Values CamPomVMLL, 29 Vegas, 60		
Pickin GamGamLL, 19 Vegas, 63 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 StoreEvent Vegas, 63 symmetrise Parameters, 46 Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
GamGamLL, 19 PrepareHadronisation GamGamLL, 20 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Vegas, 63 symmetrise Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
PrepareHadronisation symmetrise GamGamLL, 20 Parameters, 46 Process, 53 FillKinematics, 55 VXMass GetEvent, 55 GamPomVMLL, 29 IsKinematicsDefined, 55 Vegas, 60		
GamGamLL, 20 Parameters, 46 Process, 53 FillKinematics, 55 GetEvent, 55 IsKinematicsDefined, 55 Parameters, 46 VXMass GamPomVMLL, 29 Vegas, 60		
Process, 53 FillKinematics, 55 VXMass GetEvent, 55 GsKinematicsDefined, 55 VXMass GamPomVMLL, 29 Vegas, 60	•	•
FillKinematics, 55 VXMass GetEvent, 55 GamPomVMLL, 29 IsKinematicsDefined, 55 Vegas, 60		Parameters, 46
GetEvent, 55 GamPomVMLL, 29 IsKinematicsDefined, 55 Vegas, 60		\/\/\/\
IsKinematicsDefined, 55 Vegas, 60		
Setincoming Particles, 55 Generate, 62		
	Setincoming Particles, 55	Generate, 02

68 INDEX

GenerateOneEvent, 62 Integrate, 62 SetGen, 63 StoreEvent, 63 Vegas, 62