LPAIR++ 0.2

Generated by Doxygen 1.8.5

Mon Jan 6 2014 17:43:13

ii CONTENTS

# Contents

Hier	archica	l Index	1
2.1	Class I	Hierarchy	1
<b>.</b> .	<b>C</b> .		4
			1 1
5.1	Data	structures	1
Data	a Struc	ture Documentation	2
4.1	Event	Class Reference	2
	4.1.1	Detailed Description	2
	4.1.2	Member Function Documentation	2
4.2	GamG	am Class Reference	3
	4.2.1	Detailed Description	4
	4.2.2	Constructor & Destructor Documentation	5
	4.2.3	Member Function Documentation	6
4.3	GamG	amKinematics Class Reference	7
	4.3.1	Field Documentation	8
4.4	Hadro	niser Class Reference	9
	4.4.1	Detailed Description	9
4.5	Inelast	icParticle Class Reference	9
	4.5.1	Detailed Description	11
	4.5.2	Member Function Documentation	11
	4.5.3	Field Documentation	13
4.6	InputP	Parameters Class Reference	13
	4.6.1	Detailed Description	14
	4.6.2	Member Function Documentation	14
	4.6.3	Field Documentation	15
4.7	MCGe	n Class Reference	16
	4.7.1	Detailed Description	17
	4.7.2	Constructor & Destructor Documentation	17
	4.7.3	Member Function Documentation	17
4.8	Particl	e Class Reference	17
	4.8.1	Detailed Description	19
	4.8.2	Member Function Documentation	19
	4.8.3	Field Documentation	21
4.9	Proces	ss Class Reference	21
	4.9.1	Detailed Description	22
4.10	Pythia	6Hadroniser Class Reference	22
	2.1 Data 3.1 Lata 4.1 4.2 4.3 4.4 4.5 4.6 4.7	2.1 Class II  Data Struct 3.1 Data S  Data Struct 4.1 Event 4.1.1 4.1.2 4.2 GamG 4.2.1 4.2.2 4.2.3 4.3 GamG 4.3.1 4.4 Hadron 4.4.1 4.5 Inelast 4.5.1 4.5.2 4.5.3 4.6 InputP 4.6.1 4.6.2 4.6.3 4.7 MCGe 4.7.1 4.7.2 4.7.3 4.8 Particle 4.8.1 4.8.2 4.8.3 4.9 Proces 4.9.1	Data Structure Index 3.1 Data Structures  Data Structure Documentation 4.1 Event Class Reference 4.1.1 Detailed Description 4.1.2 Member Function Documentation 4.2 GamGam Class Reference 4.2.1 Detailed Description 4.2.2 Constructor & Destructor Documentation 4.2.3 Member Function Documentation 4.3 GamGamKinematics Class Reference 4.3.1 Field Documentation 4.4 Hadroniser Class Reference 4.4.1 Detailed Description 4.5 Inelastic/Particle Class Reference 4.5.1 Detailed Description 4.5.2 Member Function Documentation 4.6 InputParameters Class Reference 4.6.1 Detailed Description 4.6.3 Field Documentation 4.6 InputParameters Class Reference 4.6.1 Detailed Description 4.6.2 Member Function Documentation 4.6.3 Field Documentation 4.7 MCGen Class Reference 4.7.1 Detailed Description 4.7.2 Constructor & Destructor Documentation 4.7.3 Member Function Documentation 4.7.3 Member Function Documentation 4.8 Particle Class Reference 4.8.1 Detailed Description 4.8.2 Member Function Documentation 4.8.3 Field Documentation 4.8.3 Field Documentation 4.9 Process Class Reference

1 Todo List 1

4.11 Vegas Class Reference	22
4.11.1 Constructor & Destructor Documentation	23
4.11.2 Member Function Documentation	23
Bibliographic References	25
Index	26
1 Todo List	
<b>Global GamGam::GamGam (const unsigned int ndim_, int nOpt_, double x_[])</b> Figure out how this <i>nOpt_</i> parameter is affecting the final cross-section computation and of	events generation
2 Hierarchical Index	
2.1 Class Hierarchy	
This inheritance list is sorted roughly, but not completely, alphabetically:	
Event	2
GamGamKinematics	7
Hadroniser	9
Pythia6Hadroniser	22
InputParameters	13
MCGen	16
Particle	17
InelasticParticle	9
Process	21
GamGam	3
Vegas	22
3 Data Structure Index	
3.1 Data Structures	
Here are the data structures with brief descriptions:	
Event Kinematic information on the particles in the event	2
GamGam Computes the matrix element for a $\gamma\gamma\to\ell^+\ell^-$ process	3

GamGamKinematics	
List of kinematic cuts to apply on the central and outgoing phase space	7
Hadroniser	9
InelasticParticle	9
InputParameters	
List of input parameters used to start and run the simulation job	13
MCGen	
Core of the Monte-Carlo generator	16
Particle	
Kinematics of one particle	17
Process	21
Pythia6Hadroniser	
Pythia6 hadronisation algorithm	22
Vegas	
Vegas Monte-Carlo integrator instance	22

## 4 Data Structure Documentation

## 4.1 Event Class Reference

Kinematic information on the particles in the event.

Public Member Functions

- void Dump ()
- Particle \* GetByRole (int role\_)
- int SetParticle (Particle \*part\_)
- void Store (std::ofstream \*, double weight\_=1.)
- void StoreLHERecord (std::ofstream \*of\_, const double weight\_=1.)

Stores the LHE block for this event.

## 4.1.1 Detailed Description

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

## 4.1.2 Member Function Documentation

## 4.1.2.1 void Event::Dump ( )

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

## 4.1.2.2 **Particle**\* Event::GetByRole ( int role\_ )

Returns the pointer to the Particle object corresponding to a certain role in the process kinematics

#### **Parameters**

role_	The role the particle has to play in the process	

#### Returns

A pointer to the requested Particle object

```
4.1.2.3 int Event::SetParticle ( Particle * part_ )
```

Sets the information on one particle in the process

**Parameters** 

```
part_ The Particle object to insert or modify in the event
```

#### 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

```
4.1.2.4 void Event::Store ( std::ofstream * , double weight = 1. )
```

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

**Parameters** 

weight	The weight of the event
weight_	The Weight of the event

```
4.1.2.5 void Event::StoreLHERecord ( std::ofstream * of_, const double weight_ = 1. )
```

Stores in a LHE format (a XML-style) all the information on the particles composing this event

**Parameters** 

of_	The file stream on which the event record has to be saved
weight_	The weight of the event

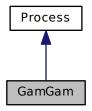
The documentation for this class was generated from the following file:

• include/event.h

## 4.2 GamGam Class Reference

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

Inheritance diagram for GamGam:



**Public Member Functions** 

GamGam (const unsigned int ndim\_, int nOpt\_, double x\_[])

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 ()

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

double ComputeWeight (int nm\_=1)

Computes the process' weight for the given point.

void FillKinematics (bool symmetrise\_=false)

Fills the Event object with the particles' kinematics.

Event \* GetEvent ()

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

- double GetT1 ()
- void GetT1extrema (double &t1min\_, double &t1max\_)
- double GetT2 ()
- void GetT2extrema (double &t2min\_, double &t2max\_)
- bool IsKinematicsDefined ()

Is the system's kinematics well defined?

bool SetIncomingKinematics (Particle ip1\_, Particle ip2\_)

Sets the momentum and PDG id for the incoming particles.

void SetKinematics (GamGamKinematics cuts\_)

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

bool SetOutgoingParticles (int part\_, int pdgld\_)

Sets the PDG id for the outgoing particles.

## 4.2.1 Detailed Description

Full class of methods and objects to compute the full analytic matrix element [2] for the  $\gamma\gamma \to \ell^+\ell^-$  process according to a set of kinematic constraints provided for the incoming and outgoing particles (the GamGam-Kinematics object).

- 4.2.2 Constructor & Destructor Documentation
- 4.2.2.1 GamGam::GamGam ( const unsigned int ndim\_, int nOpt\_, double  $x_{[]}$  )

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

#### **Parameters**

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

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

4.2.3 Member Function Documentation

4.2.3.1 void GamGam::ComputeCMenergy (

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

4.2.3.2 double GamGam::ComputeMX ( double x\_, double outmass\_, double \* dw\_ )

Computes the mass of the outgoing proton remnant if any

**Parameters** 

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

#### Returns

The mass of the outgoing proton remnant

4.2.3.3 double GamGam::ComputeWeight ( int nm\_ = 1 )

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

Returns

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

4.2.3.4 void GamGam::FillKinematics ( bool symmetrise\_ = false )

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

4.2.3.5 **Event**\* GamGam::GetEvent() [inline]

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

4.2.3.6 double GamGam::GetT1 ( ) [inline]

Returns the value for the first photon virtuality

Returns

 $t_1$ , the first photon virtuality

4.2.3.7 void GamGam::GetTlextrema ( double & tlmin\_, double & tlmax\_ ) [inline]

Returns the two limit values for the first photon virtuality

#### **Parameters**

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

4.2.3.8 double GamGam::GetT2 ( ) [inline]

Returns the value for the second photon virtuality

Returns

 $t_2$ , the second photon virtuality

4.2.3.9 void GamGam::GetT2extrema ( double & t2min\_, double & t2max\_ ) [inline]

Returns the two limit values for the second photon virtuality

#### **Parameters**

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

4.2.3.10 bool GamGam::IsKinematicsDefined ( ) [inline]

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

4.2.3.11 bool GamGam::SetIncomingKinematics ( Particle ip1\_, Particle ip2\_ )

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

## **Parameters**

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

4.2.3.12 void GamGam::SetKinematics ( GamGamKinematics cuts\_ )

## **Parameters**

cuts	The Cuts object containing the kinematic parameters

4.2.3.13 bool GamGam::SetOutgoingParticles ( int part\_, int pdgld\_ )

#### **Parameters**

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

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

• include/gamgam.h

## 4.3 GamGamKinematics Class Reference

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

B CONTENTS

#### 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 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 (  $\theta_{\min}$ ) 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.

#### 4.3.1 Field Documentation

#### 4.3.1.1 int GamGamKinematics::kinematics

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

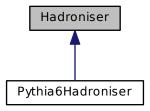
- 0 for the electron-electron elastic case
- 1 for the proton-proton elastic case
- 2 for the proton-proton single-dissociative (or inelastic) case
- 3 for the proton-proton double-dissociative case

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

include/gamgam.h

## 4.4 Hadroniser Class Reference

Inheritance diagram for Hadroniser:



Public Member Functions

bool Hadronise (Particle \*part\_)
 Main caller to hadronise a particle.

## 4.4.1 Detailed Description

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

Author

 ${\tt Laurent\ Forthomme\ laurent.forthomme\ @uclouvain.be}$ 

Date

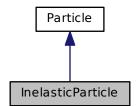
January 2014

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

• include/hadroniser.h

## 4.5 InelasticParticle Class Reference

Inheritance diagram for InelasticParticle:



**Public Member Functions** 

void 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.

Particle \* GetDaughter (const unsigned int num\_=0)

Gets a daughter from this particle, labelled by its identifier in this particle's daughters list.

- std::string GetLHEline (bool revert\_=false)
- Particle \* GetMother ()

Gets 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 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.

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

double eta

Pseudo-rapidity.

double p

Norm of the 3-momentum, in GeV/c.

int pdgld

Particle Data Group integer identifier.

double pt

Transverse momentum, in GeV/c.

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.

#### 4.5.1 Detailed Description

Class containing the information on a particle supposed to decay or fragment in the process

4.5.2 Member Function Documentation

4.5.2.1 void Particle::AddDaughter ( Particle \* part\_ ) [inherited]

**Parameters** 

part\_ The Particle object in which this particle will desintegrate or convert

4.5.2.2 void Particle::Dump ( ) [inherited]

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

4.5.2.3 void Particle::E ( double E\_ ) [inline], [inherited]

**Parameters** 

E\_ Energy, in GeV

4.5.2.4 **Particle**\* Particle::GetDaughter ( const unsigned int num\_ = 0 ) [inherited]

Returns

A Particle object containing all the kinematic information related to this daughter particle

4.5.2.5 std::string Particle::GetLHEline ( bool revert\_ = false ) [inherited]

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

**Parameters** 

*revert\_* Is the event symmetric? If set to true, the third component of the momentum is reverted.

Returns

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

4.5.2.6 bool InelasticParticle::Hadronise ( std::string algo\_ )

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

algo\_ | Algorithm in use to hadronise the particle

4.5.2.7 double Particle::M ( ) [inline], [inherited]

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

## Returns

The particle's mass

4.5.2.8 bool Particle::P ( double px\_, double py\_, double pz\_ ) [inline], [inherited]

#### **Parameters**

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

Here is the call graph for this function:



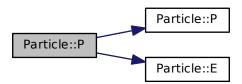
 $4.5.2.9 \quad bool \; Particle::P \; \big( \; \; double \; px\_, \; \; double \; py\_, \; \; double \; pz\_, \; \; double \; E\_ \; \big) \quad [inline], \; [inherited]$ 

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

## Parameters

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

Here is the call graph for this function:



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

#### **Parameters**

<i>p</i>	3-momentum
E_	Energy, in GeV

4.5.2.11 void Particle::SetMother ( Particle \* part\_ ) [inline], [inherited]

**Parameters** 

part\_ | A Particle object containing all the information on the mother particle

#### 4.5.3 Field Documentation

## 4.5.3.1 int Particle::status [inherited]

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

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

■ include/inelastic.h

## 4.6 InputParameters Class Reference

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

**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\_)

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.

Stores the full run configuration to an external config file.

bool generation

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

• int gpdf

PDFLIB group to use.

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.

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 maxtheta

Maximal polar angle  $\theta$  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  ${\cal M}_X$  of the outgoing proton remnants.

double minpt

Minimal  $p_T$  of the outgoing leptons.

double mintheta

Minimal polar angle  $\theta$  of the outgoing leptons.

• int ngen

Number of events already generated in this run.

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.

• 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.

#### 4.6.1 Detailed Description

Note

The default parameters are derived from GMUINI in LPAIR

## 4.6.2 Member Function Documentation

## 4.6.2.1 bool InputParameters::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**

inFile_	Name of the configuration file to load

4.6.2.2 void InputParameters::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  $\theta$ , the polar angle.

#### **Parameters**

etamin_	The minimal value of $\eta$ for the outgoing leptons
etamax_	The maximal value of $\eta$ for the outgoing leptons

4.6.2.3 bool InputParameters::StoreConfigFile ( std::string outFile\_ )

#### **Parameters**

	Nicolar Calacter Caracter Classic Constitution
outFile	Name of the configuration file to create
0411 110_	Traine of the comparation me to create

- 4.6.3 Field Documentation
- 4.6.3.1 bool InputParameters::debug

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

4.6.3.2 double InputParameters::maxmx

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

4.6.3.3 double InputParameters::maxpt

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

4.6.3.4 int InputParameters::mcut

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

- 0 No cuts at all (for the total cross section)
- 1 Vermaserens' hypothetical detector cuts : for both leptons,
  - $\frac{|p_z|}{|\mathbf{p}|} \leq$  0.75 and  $p_T \geq 1$  GeV/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
- 4.6.3.5 double InputParameters::minmx

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

4.6.3.6 double InputParameters::minpt

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

4.6.3.7 int InputParameters::ntreat

Note

Is it correctly implemented?

#### 4.6.3.8 int InputParameters::p1mod

The first incoming particle type and kind of interaction :

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

Note

Was named PMOD in II PAIR

4.6.3.9 int InputParameters::p2mod

Note

Was named EMOD in ILPAIR

4.6.3.10 int InputParameters::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

## 4.6.3.11 bool InputParameters::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!

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

• include/utils.h

## 4.7 MCGen Class Reference

Core of the Monte-Carlo generator.

**Public Member Functions** 

- MCGen (InputParameters ip\_)
  - Class constructor.
- void ComputeXsection (double \*, double \*)
- InputParameters GetInputParameters ()

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

## 4.7.1 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 GamGam 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

- 4.7.2 Constructor & Destructor Documentation
- 4.7.2.1 MCGen::MCGen ( InputParameters ip\_ )

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

Parameters

*ip\_* List of input parameters defining the phase space on which to perform the integration

- 4.7.3 Member Function Documentation
- 4.7.3.1 void MCGen::ComputeXsection ( double \* , double \* )

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

4.7.3.2 InputParameters MCGen::GetInputParameters ( ) [inline]

Returns

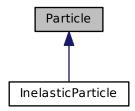
The InputParameter object embedded in this class

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

- include/mcgen.h
- 4.8 Particle Class Reference

Kinematics of one particle.

Inheritance diagram for Particle:



**Public Member Functions** 

```
void 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.

Particle \* GetDaughter (const unsigned int num\_=0)

Gets a daughter from this particle, labelled by its identifier in this particle's daughters list.

- std::string GetLHEline (bool revert\_=false)
- Particle \* GetMother ()

Gets the mother particle from which this particle arises.

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 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.

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

double eta

Pseudo-rapidity.

double p

Norm of the 3-momentum, in GeV/c.

int pdgld

Particle Data Group integer identifier.

double pt

Transverse momentum, in GeV/c.

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.

#### 4.8.1 Detailed Description

Kinematic information for one particle

4.8.2 Member Function Documentation

4.8.2.1 void Particle::AddDaughter ( Particle \* part\_ )

Parameters

part\_ The Particle object in which this particle will desintegrate or convert

4.8.2.2 void Particle::Dump (

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

4.8.2.3 void Particle::E ( double E\_ ) [inline]

**Parameters** 

E\_ Energy, in GeV

4.8.2.4 **Particle**\* Particle::GetDaughter ( const unsigned int num $_{-}$  = 0 )

Returns

A Particle object containing all the kinematic information related to this daughter particle

4.8.2.5 std::string Particle::GetLHEline ( bool revert\_ = false )

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

#### **Parameters**

## Returns

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

4.8.2.6 double Particle::M ( ) [inline]

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

## Returns

The particle's mass

4.8.2.7 bool Particle::P ( double px\_, double py\_, double pz\_ ) [inline]

#### **Parameters**

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

Here is the call graph for this function:



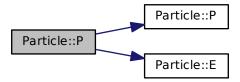
4.8.2.8 bool Particle::P ( double px\_, double py\_, double pz\_, double E\_ ) [inline]

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

## Parameters

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

Here is the call graph for this function:



4.8.2.9 bool Particle::P ( double p\_[3], double E\_ )

## Parameters

<i>p</i>	3-momentum
E_	Energy, in GeV

4.8.2.10 void Particle::SetMother ( Particle \* part\_ ) [inline]

#### **Parameters**

part\_ A Particle object containing all the information on the mother particle

#### 4.8.3 Field Documentation

#### 4.8.3.1 int Particle::status

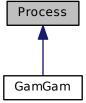
Codes 1-10 correspond to currently existing partons/particles, and larger codes contain partons/particles which no longer exist, or other kinds of event information

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

• include/particle.h

## 4.9 Process Class Reference

Inheritance diagram for Process:



Public Member Functions

double ComputeWeight ()

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

## 4.9.1 Detailed Description

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

Author

Laurent Forthomme laurent.forthomme@uclouvain.be

Date

January 2014

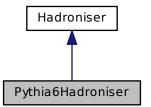
The documentation for this class was generated from the following file:

include/process.h

## 4.10 Pythia6Hadroniser Class Reference

Pythia6 hadronisation algorithm.

Inheritance diagram for Pythia6Hadroniser:



Public Member Functions

bool Hadronise (Particle \*part\_)
 Main caller to hadronise a particle.

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

• include/pythia6hadroniser.h

## 4.11 Vegas Class Reference

Vegas Monte-Carlo integrator instance.

**Public Member Functions** 

- Vegas (const int dim\_, double f\_(double \*, size\_t, void \*), InputParameters \*inParam\_)
- ~Vegas ()

Class destructor.

• int Integrate (double \*result\_, double \*abserr\_)

Launches the integration of the provided function.

• int LaunchGeneration ()

Launches the generation of events.

• int MyIntegrate (double \*result\_, double \*abserr\_)

#### 4.11.1 Constructor & Destructor Documentation

```
4.11.1.1 Vegas::Vegas ( const int dim_, double f_double *, size_t, void *, InputParameters * inParam_ )
```

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

#### **Parameters**

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

#### 4.11.2 Member Function Documentation

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

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

#### **Parameters**

result_	The cross section as integrated by Vegas for the given phase space restrictions	
abserr_	The error associated to the computed cross section	

```
4.11.2.2 int Vegas::LaunchGeneration ( )
```

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

```
4.11.2.3 int Vegas::MyIntegrate ( double * result_, double * abserr_ )
```

Vegas algorithm to perform the (\_dim)-dimensional Monte Carlo integration of a given function as described in [1]

## 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

result_	The cross section as integrated by Vegas for the given phase space restrictions	
abserr_	The error associated to the computed cross section	

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

include/vegas.h

REFERENCES 25

# References

[1] G Peter Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192 – 203, 1978. 23

[2] J.A.M. Vermaseren. Two-photon processes at very high energies. *Nuclear Physics B*, 229(2):347 – 371, 1983. 4

# Index

Compute CMenergy GamGam, 6 Compute MX GamGam, 6 Compute MX GamGam, 6 Compute Weight GamGam, 6 Compute Weight GamGam, 6 Compute Weight GamGam, 6 Compute Weight GamGam, 6 GetT1 GamGam, 6 GetT2 GamGam, 6 GetT2 GamGam, 7  InputParameters, 15 Dump Event, 2 Inelastic Particle, 11 Particle, 19 E Inelastic Particle, 11 Particle, 19 Event, 2 Dump, 2 GetByRole, 2 SetParticle, 3 Store, 15 Store, 16 Store, 17 Store, 16 Store, 16 Store, 17 St	AddDaughter InelasticParticle, 11 Particle, 19	GetEvent GamGam, 6 GetInputParameters
GamGam, 6 ComputeWeight GamGam, 6 ComputeXsection MCGen, 17 GetTlextrema GamGam, 6 GetTlextrema GamGam, 6 GetTlextrema GamGam, 6 GetTlextrema GamGam, 6 GetTlextrema GamGam, 7 GetTextrema GamGam, 7  Bettent, 2 InelasticParticle, 11 Particle, 19 Event, 2 InelasticParticle, 11 Particle, 19 Event, 2 Dump, 2 GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 Store, 3 Store, 3 Store, 14 FillKinematics GamGam, 6 GamGam, 3 ComputeCMenergy, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1, 6 GetT1extrema, 6 GetT1, 6 GetT1extrema, 7 Iskinematics, 7 SetCoungingParticles, 7 GetDaughter InelasticParticle, 11 GetDaughter InelasticParticle, 11 InelasticParticle	GamGam, 6	
GamGam, 6 ComputeXsection MCGen, 17 GetT2 GamGam, 7  Dump Event, 2 InelasticParticle, 11 Particle, 19  EnelasticParticle, 11 Particle, 19  EnelasticParticle, 11 Particle, 19  Event, 2 InelasticParticle, 11 Particle, 19  Event, 2 Dump, 2 GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 Store, 3 Store, 3 StoreLHERecord, 3 FillKinematics GamGam, 6  GamGam, 6  GamGam, 6  GamGam, 3 ComputeCMenergy, 6 ComputeMX, 6 ComputeMX, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GatEvent, 6 GetT1, 6 GetT1extrema, 6 GetT2, 7 GetT2extrema, 7 IskinematicsDefined, 7 SetHcomingKinematics, 7 SetCoungingKinematics, 7 SetNoremingKinematics, 7 SetNoremingKinematics, 7 SetNoremingKinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11  GetT1extrema GamGam, 7 GetT2extrema GamGam, 6 GetT1extrema GamGam, 7 Integrate GamGam, 7 Integrate Vegas, 23 IskinematicsDefined GamGam, 7 Iskinematics, 8 GetBughter InelasticParticle, 11  LaunchGeneration	GamGam, 6	GetT1
MCGen, 17  debug InputParameters, 15 Dump Event, 2 InelasticParticle, 11 Particle, 19  Event, 2 Jump, 2 GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 Store, 3 Store, 4 SetMother, 13 StoreLHERecord, 3  FillKinematics GamGam, 5 GamGam, 5 GamGam, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GamGam, 5 GamGam, 5 GamGam, 6 GetT1extrema, 6 GetT1extrema, 6 GetT2, 7 GetT2 GattParticle, 19 Hadronise InelasticParticle, 11 Hadroniser, 9  InelasticParticle, 11 Dump, 11 E, 11 Dump, 11 E, 11 Dump, 11 E, 11 GetLHEline, 11 Hadronise, 11 Hadronise, 11 Hadronise, 11 Status, 13 InputParameters, 13 debug, 15 maxmx, 15 maxmx, 15 maxmx, 15 maxmx, 15 minmx, 15	GamGam, 6	GetT1extrema
debug InputParameters, 15 Dump Event, 2 InelasticParticle, 11 Particle, 19  E InelasticParticle, 11 Particle, 19  E InelasticParticle, 11 Particle, 19  E InelasticParticle, 11 Particle, 19  Event, 2 InelasticParticle, 11 Particle, 19  Event, 2 Dump, 2 GetByRole, 2 SetParticle, 3 Store, 3 StoreLHERecord, 3 Store, 3 StoreLHERecord, 3 FillKinematics GamGam, 6  GamGam, 6  GamGam, 6  GamGam, 5 ComputeVMeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GamGam, 5 GetTevent, 6 GetT1, 6 GetT2, 7 GetT2extrema, 7 IskinematicsDefined, 7 SetMinematics, 7 Kinematics, 8 GetBugNole Event, 2 GetDaughter InelasticParticle, 11  Hadronise Inel	•	GetT2
Dump Event, 2 InelasticParticle, 11 Particle, 19  E InelasticParticle, 11 Particle, 19  E InelasticParticle, 11 Particle, 19  E InelasticParticle, 11 Particle, 19  Event, 2 Dump, 2 GetByRole, 2 GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 Store, 3 Store, 3 StoreHERecord, 3  FillKinematics GamGam, 6 InputParameters, 13 Gebug, 15 maxmx, 15 maxmx, 15 maxmx, 15 maxmx, 15 minmx, 15 minmx, 15 GamGam, 5 GamGam, 5 GetWent, 6 GetTlextrema, 6 GetTlextrema, 6 GetT1extrema, 6 GetT1extrema, 7 IsKinematics, 7 SetIncomingKinematics, 7 SetIncomingKinematics, 7 SetNinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 7 Kinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11  Hadronise InelasticParticle, 11  Hadronise, 1 InelasticParticle, 11  Hadronise, 9  AddDaughter, 11  Dump, 11  Lanchaddau, 11  Hadronise, 11  Had	_	GetT2extrema
Particle, 19  E InelasticParticle, 11 Particle, 19 Event, 2 Dump, 2 GetByRole, 2 SetParticle, 3 Store, 3 StoreLHERecord, 3 FillKinematics GamGam, 6 ComputeCMenergy, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1, 6 GetT2extrema, 7 IsKinematics, 7 SetRolinematics, 7 SetRolinematics, 7 SetRolinematics, 7 SetRolinematics, 7 SetRolinematics, 8 GetByRole Event, 2 GetDaughter, 11 Dump, 11 Dubles 10  EvetDaughter, 10  Event, 2 DetDaughter, 10  Event, 2	Dump	
InelasticParticle, 11 Particle, 19 Event, 2 Dump, 2 GetByRole, 2 GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 StoreLHERecord, 3 FillKinematics GamGam, 6 GomputeCMenergy, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GetEvent, 6 GetEvent, 6 GetT1, 6 GetT1extrema, 6 GetT1extrema, 7 Iskinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 7 Kinematics, 8 GetByRole Event, 2 GetDaughter, 11 Dump, 11 E, 12 E, 12 E, 11 Dump, 11 E, 12 E, 12 E, 12 ExtMenoinse, 11 E, 12 ExtMenoinse, 11 E, 12 ExtMenoinse, 11 E, 12 ExtMenoinse, 11 Extensionse, 11 Extensionse, 12 ExtMenoinse, 12 ExtMenoinse, 12 ExtMenoinse, 13 ExtMenoinse, 14 ExtMenoinse, 14 ExtMenoinse, 15 ExtMenoinse, 15 ExtMenoinse, 16 Integrate Vegas, 23 IsKinematicsDefined GamGam, 7 Kinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11 LaunchGeneration		
Event, 2 Dump, 2 GetByRole, 2 GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 StoreLHERecord, 3 FillKinematics GamGam, 6 GamGam, 6 GamGam, 3 ComputeCMenergy, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1, 6 GetT1, 6 GetT12xtrema, 6 GetT2, 7 GetT2extrema, 7 IsKinematics, 7 SetNomingKinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 8 GetByRole Event, 2 GetDaughter, 11 GetDaught	InelasticParticle, 11	AddDaughter, 11
GetByRole, 2 SetParticle, 3 Store, 3 Store, 3 StoreLHERecord, 3 StoreLHERecord, 3 StoreLHERecord, 3 FillKinematics GamGam, 6 GamGam, 6 InputParameters, 13 debug, 15 maxmx, 15 ComputeCMenergy, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1, 6 GetT1, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetWinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 7 Kinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11  GetTamCarter InelasticParticle, 11  GetLHEline, 11 Hadronise, 11 Hatronice, 11 Hadronie, 12 Hadronie, 12 Hadronie, 12	Event, 2	E, 11
Store, 3 StoreLHERecord, 3 P, 12 SetMother, 13 Status, 13 InputParameters, 13 debug, 15 maxmx, 15 ComputeCMenergy, 6 ComputeMX, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1, 6 GetT1extrema, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11  M, 11 P, 12 SetMother, 13 SetMother, 14 SetBange, 15 StoreConfigFile, 14 SetEtaRange, 15 StoreConfigFile, 15 symmetrise, 16 Integrate Vegas, 23 IsKinematicsDefined GamGam, 7 kinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11 LaunchGeneration	GetByRole, 2	GetLHEline, 11
FillKinematics GamGam, 6  GamGam, 6  InputParameters, 13 debug, 15 maxmx, 15 ComputeCMenergy, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetOutgoingParticles, 7  GamGamKinematics, 8  GetByRole Event, 2 GetDaughter InelasticParticle, 11  InputParameters, 13 status, 13 InputParameters, 13 debug, 15 maxmx, 15 m	Store, 3	M, 11
GamGam, 6  GamGam, 3  ComputeCMenergy, 6  ComputeMX, 6  ComputeWeight, 6  FillKinematics, 6  GamGam, 5  GetEvent, 6  GetT1, 6  GetT1extrema, 6  GetT2extrema, 7  IsKinematicsDefined, 7  SetIncomingKinematics, 7  SetSuncomingKinematics, 7  Kinematics, 8  GetByRole  Event, 2  GetDaughter  InelasticParticle, 11  Integrate  LaunchGeneration  InputParameters, 13  debug, 15  maxmx, 15  minut, 15  maxmx, 15  minut, 15  minut, 15  min		SetMother, 13
ComputeCMenergy, 6 ComputeMX, 6 ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetVoutgoingParticles, 7 GamGamKinematics, 8 GetDaughter InelasticParticle, 11 Indiatic Mark 15 maxpt, 15 maxpt, 15 mcut, 15 mcut, 15 mcut, 15 minmx, 15 minmy minmy minmx, 15 minmy minmy minmx minmy minmy minmx minmy		InputParameters, 13
ComputeWeight, 6 FillKinematics, 6 GamGam, 5 GamGam, 5 GamGam, 5 GetEvent, 6 GetT1, 6 GetT1extrema, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11 Integrate  minmx, 15 minmx, 16 minmx, 15 minmx, 16 minmx, 15 minmx, 16 minmx, 15 minmx, 15 minmx, 15 minmx, 15 minmx, 16 minmx, 15 min	ComputeCMenergy, 6	
GamGam, 5 GamGam, 5 GatEvent, 6 GetEvent, 6 GetT1, 6 GetT1, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetOutgoingParticles, 7 GetByRole Event, 2 GetDaughter InelasticParticle, 11 Integrate Inte	ComputeWeight, 6	minmx, 15
GetEvent, 6 GetT1, 6 GetT1, 6 GetT1extrema, 6 GetT2, 7 GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetVinematics, 7 SetOutgoingParticles, 7 GetByRole Event, 2 GetDaughter InelasticParticle, 11  ppmod, 16 pair, 16 ReadConfigFile, 14 SetEtaRange, 15 StoreConfigFile, 15 symmetrise, 16 Integrate Vegas, 23 IsKinematicsDefined GamGam, 7 kinematics, 8  GetByRole Event, 2 GamGamKinematics, 8  LaunchGeneration	GamGam, 5	ntreat, 15
GetT1extrema, 6  GetT2, 7  GetT2extrema, 7  IsKinematicsDefined, 7  SetIncomingKinematics, 7  SetKinematics, 7  SetOutgoingParticles, 7  GamGamKinematics, 7  kinematics, 8  GetByRole  Event, 2  GetDaughter  InelasticParticle, 11  ReadConfigFile, 14  SetEtaRange, 15  StoreConfigFile, 15  symmetrise, 16  Integrate  Vegas, 23  IsKinematicsDefined  GamGam, 7  kinematics, 8  GamGam, 7  kinematics  GamGamKinematics, 8  LaunchGeneration	GetEvent, 6	p2mod, 16
GetT2extrema, 7 IsKinematicsDefined, 7 SetIncomingKinematics, 7 SetKinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 7 kinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11  StoreConfigFile, 15 symmetrise, 16 Integrate Vegas, 23 IsKinematicsDefined GamGam, 7 GamGam, 7 kinematics, 8  Kinematics GamGamKinematics, 8  LaunchGeneration	GetT1extrema, 6	ReadConfigFile, 14
SetIncomingKinematics, 7 SetKinematics, 7 SetOutgoingParticles, 7 GamGamKinematics, 7 kinematics, 8 GetByRole Event, 2 GetDaughter InelasticParticle, 11 Integrate Vegas, 23 IsKinematicsDefined GamGam, 7 kinematics GamGam, 7 kinematics GamGamKinematics, 8 LaunchGeneration	GetT2extrema, 7	StoreConfigFile, 15
SetOutgoingParticles, 7  GamGamKinematics, 7  kinematics, 8  GetByRole Event, 2  GetDaughter InelasticParticle, 11  IsKinematicsDefined  GamGam, 7  kinematics  Kinematics  GamGamKinematics, 8  LaunchGeneration	SetIncomingKinematics, 7	Integrate
kinematics, 8  GetByRole kinematics Event, 2 GamGamKinematics, 8  GetDaughter InelasticParticle, 11 LaunchGeneration		IsKinematicsDefined
GetDaughter InelasticParticle, 11 LaunchGeneration		·
	GetDaughter	

INDEX 27

M	GamGam, 7
InelasticParticle, 11	SetParticle
Particle, 20	Event, 3
MCGen, 16	status
ComputeXsection, 17	InelasticParticle, 13
GetInputParameters, 17	Particle, 21
MCGen, 17	Store
MCGen, 17	Event, 3
maxmx	${\sf StoreConfigFile}$
InputParameters, 15	InputParameters, 15
maxpt	StoreLHERecord
InputParameters, 15	Event, 3
mcut	symmetrise
InputParameters, 15	InputParameters, 16
minmx	
InputParameters, 15	Vegas, 22
minpt	Integrate, 23
InputParameters, 15	LaunchGeneration, 23
MyIntegrate	MyIntegrate, 23
Vegas, 23	Vegas, 23
ntreat	
InputParameters, 15	
•	
P	
InelasticParticle, 12	
Particle, 20, 21	
p1mod	
InputParameters, 16	
p2mod	
InputParameters, 16	
pair	
InputParameters, 16	
Particle, 17	
AddDaughter, 19	
Dump, 19	
E, 19	
GetDaughter, 19	
GetLHEline, 19	
M, 20	
P, 20, 21	
SetMother, 21	
status, 21	
Process, 21	
Pythia6Hadroniser, 22	
, ,	
ReadConfigFile	
InputParameters, 14	
SetEtaRange	
InputParameters, 15	
SetIncomingKinematics	
GamGam, 7	
SetKinematics	
GamGam, 7	
SetMother	
InelasticParticle, 13	
Particle, 21	
SetOutgoingParticles	