The Jpp - JPhysics package

M. de Jong

December 1, 2021

Abstract

The Jpp - JPhysics package is a set of C^{++} methods, interfaces and classes that can be used to calculate the probability density functions of the arrival time of Cherenkov light.

1 Introduction

The probability density functions (PDFs) of the arrival time of Cherenkov light have been worked out in reference [1]. This note describes the C^{++} methods, interfaces and classes that can be used to calculate these PDFs. Auxiliary classes for the numerical computation of integrals are also described.

1.1 Global methods

The list of global methods that are included in the Jpp - JPhysics package are summarised in table 1. In this, the wavelength should be expressed in nm, the energy in GeV and the distance in m.
The method `getIndexOfRefraction()` refers to some average index of refraction, n, that is used to determine the offset, t_0, of the arrival time of light.

$$t_0 \equiv \frac{R \tan\theta_C}{c} \tag{1}$$

where R refers to distance of closest approach of the muon track to the PMT, c to the speed of light in vacuum and $\tan\theta_C = \sqrt{(1-n)\times(1+n)}$.

1.2 Implementation of PDFs

The interface, JPDF, is used to define the way properties of the PMT and the medium that are needed for the determination of the PDFs can be specified. The interface provides for the implementations of the equations of the PDFs in reference [1] In this, the wavelength should be expressed in nm, the energy in GeV and the distance in m.

```
class JPDF :
  public virtual JDispersionInterface,
  public virtual JAbstractPMT,
  public virtual JAbstractMedium

{
  JPDF(const double Wmin,
       const double Wmax,
```

method	result
`getIndexOfRefraction()`	index of refraction of the medium
`getTanThetaC()`	$\tan\theta_C$
`cherenkov(...)`	Number of photons per unit track length and per unit wavelength as a function of wavelength and index of refraction [4]
`geanc()`	Equivalent unit track length per unit shower energy [1]
`gWater(...)`	Energy of muon in water as a function of its initial energy and traversed distance [6]
	Range of muon as a function of energy [6]
	Equivalent unit track length per unit shower energy and per unit track length [1]
`gRock(...)`	Idem for rock
`geant(...)`	Probability of photon emission from EM-showers per unit solid angle as a function of index of refraction and emission angle [5]
`geanz(...)`	Probability of photon emission from EM-showers per unit length as a function of energy and distance from vertex [7]

Table 1: List of global methods.

```
      const int    numberOfPoints = 20,
      const double epsilon        = 1e-12)
  {}

  double getDirectLightFromMuon(...)         const;
  double getDirectLightFromEMshowers(...)    const;
  double getDirectLightFromDeltaRays(...)    const;
  double getDirectLightFromEMshower(...)     const;
  double getScatteredLightFromMuon(...)      const;
  double getScatteredLightFromEMshowers(...) const;
  double getScatteredLightFromDeltaRays(...) const;
  double getScatteredLightFromEMshower(...)  const;
};
```

The arguments to the constructor are the minimal wavelength, the maximal wavelength and the number of points for the integration, respectively. The optional fourth argument refers to the precision of the evaluation of the abscissa values. The `JGaussLegendre` class is used to evaluate the integrals required for the determination of the PDFs. The implementations of the equations of the PDFs in reference [1] are listed in table 2. The arguments to these methods are the distance of closest approach of the muon track to the PMT, the zenith and azimuth angle of the orientation of the PMT and the arrival time of light, respectively. The coordinate system is defined such that the muon travels along the $z-$axis and the position of the PMT is located in the $x-z$ plane [1]. The arrival time is defined relative to the t_0 from equation 1. For Cherenkov light from the muon, the PDF is expressed as the probability per unit time. For light from Electro-Magnetic showers, the PDF is expressed as the probability per unit time and per unit energy. It has

been shown that, to good approximation, the total amount of light due to Electro-Magnetic showers corresponding to the energy loss of the muon is proportional to the energy of the muon [1].

method	equation
`getDirectLightFromMuon(...)`	43
`getDirectLightFromEMshower(...)`	44
`getDirectLightFromDeltaRays(...)`	47
`getDirectLightFromEMshowers(...)`	47
`getScatteredLightFromMuon(...)`	48
`getScatteredLightFromEMshower(...)`	49
`getScatteredLightFromDeltaRays(...)`	50
`getScatteredLightFromEMshowers(...)`	50

Table 2: List of member methods of the `JPDF` interface and the corresponding equations in reference [1].

The `JPDF` class derives from three interfaces that describe the features of the PMT, the medium and the light propagation, respectively.

The `JAbstractPMT` interface describes the key features of the PMT, namely:

```
class JAbstractPMT {
  virtual double getPhotocathodeArea()              const = 0;
  virtual double getQE(const double)                const = 0;
  virtual double getAngularAcceptance(const double) const = 0;
};
```

The `JAbstractMedium` interface describes the key features of the medium, namely:

```
class JAbstractMedium {
  virtual double getAbsorptionLength(const double)      const = 0;
  virtual double getScatteringLength(const double)      const = 0;
  virtual double getScatteringProbability(const double) const = 0;
};
```

The `JDispersionInterface` interface describes the key features of the light propagation, namely:

```
class JAbstractMedium {
  virtual double getIndexOfRefractionPhase(const double) const = 0;
  virtual double getDispersionPhase(const double)        const = 0;
  virtual double getIndexOfRefractionGroup(const double) const = 0;
  virtual double getDispersionGroup(const double)        const = 0;
};
```

The `JDispersion` class provides for the implementation of the dispersion relations for water in the deep-sea [8].

```
class JDispersion :
  public virtual JDispersionInterface
{
  JDispersion(const double P_atm)
  {}

  double getIndexOfRefractionPhase(const double&) const;
  double getDispersionPhase(const double&)        const;
  double getIndexOfRefractionGroup(const double&) const;
  double getDispersionGroup(const double&)        const;
};
```

The dispersion of light depends on the pressure of the water. The first argument to the constructor can be used to specify the pressure (c.q. depth) of the water (unit Atm).

In order to take the features of the PMT and the medium into account, a designed class should be written that is derived from the JPDF interface. This class should then provide for implementations of the corresponding (i.e. virtual) methods. To this end, a simple wrapper class, JPDF_C, is available that can be used to pass the information in the form of global methods (i.e. C-like).

```
class JPDF_C : public JAbstractPDF
{
  JPDF_C(const double,
         //
         // pointers to global methods
         //
         double (*) (const double),
         double (*) (const double),
         double (*) (const double),
         double (*) (const double),
         double (*) (const double),
         //
         // parameters for base class
         //
         const double P_atm,
         const double Wmin,
         const double Wmax,
         const int    numberOfPoints,
         const double epsilon)
  {}
};
```

The first argument to the constructor refers to the photo-cathode area of the PMT. The following arguments are pointers to global methods corresponding to *i)* the quantum efficiency of PMT as a function of the wavelength, *ii)* the angular acceptance of PMT as a function of the angle of incidence, *iii)* the absorption length as a function of the wavelength, *iv)* the scattering length as a function of the wavelength and *v)* the angular distribution of light scattering, respectively. The last four arguments have the same meaning as those of the JPDF constructor.

2 Applications

The list of applications that are available in the Jpp framework includes JDrawPDF, JDrawPDG, JMakePDF, JMakePDG, JMakeCDF, JMakeCDG, JBlurPDF, JPlotPDF, JPlotPDG and JDemoPDF. These programs are briefly described in the following.

The JDrawPDF and JDrawPDG programs can be used to create a histogram of the PDF of light from a muon and an EM-shower, respectively. The list of command line options includes:

option	parameter
-o	output file name
-R	distance of closest approach of the track to the PMT
-D	zenith and azimuth angle of the orientation of the PMT
-c	cosine angle EM shower direction and EM shower - PMT position
-F	function type

The output file is in ROOT format [9]. The two orientation angles of the PMT should be surrounded by quotes to ensure they are parsed together. The possible values of the function type are listed in table 3. The option -c is only valid for program JDrawPDG.

The JMakePDF and JMakePDG programs can be used to create a multi-dimensional table of the PDF of light from a muon and an EM-shower, respectively. The list of command line options includes:

option	parameter
-o	output file name
-n	number of points for integration
-e	precision of the evaluation of the abscissa values
-F	function type

The output file contains the binary data of the tabulated PDF. The default values for the number of points for integration and the precision of the evaluation of the abscissa values yield a reasonable numerical accuracy for the function value interpolation. The possible values of the function type are listed in table 3.

The JMakeCDF and JMakeCDG programs can be used to convert the multi-dimensional tables of the PDF to multi-dimensional tables of the cumulative density functions (CDFs). The CDFs can be used in turn to generate efficiently random data according to the PDFs. The list of command line options includes:

option	parameter
-f	input file name
-o	output file name

The output file contains the binary data of the tabulated CDF.

The JBlurPDF program can be used to take into account the effect of the transition time spread (TTS) of the P)MT on the PDF. This effect is calculated numerically by the convolution of the PDF and a Gaussian function with a $\sigma = \mathrm{TTS}$. The list of command line options includes:

option	parameter
-f	input file name
-o	output file name
-n	number of points for integration
-e	precision of the evaluation of the abscissa values
-T	TTS [ns]

The input file contains the binary data of the tabulated PDF. The output file contains the binary data of the newly tabulated PDF. The number of points for the integration and the precision of the evaluation of the abscissa values apply to the `JGaussHermite` integration. The default values yield a reasonable numerical accuracy.

The `JPlotPDF` and `JPlotPDG` programs can be used to create a histogram of the PDF based on interpolation of the tables with function values created with the `JMakePDF` and `JMakePDG` programs, respectively. The list of command line options includes:

option	parameter
`-f`	input file name
`-o`	output file name
`-R`	distance of closest approach of the track to the PMT
`-D`	zenith and azimuth angle of the orientation of the PMT
`-c`	cosine angle EM shower direction and EM shower - PMT position

The input file contains the binary data of the tabulated PDF. The output file is in ROOT format [9]. The two orientation angles of the PMT should be surrounded by quotes to ensure they are parsed together. The option `-c` is only valid for program `JPlotPDG`.

The `JDemoPDF` program serves as an example for using the PDFs in a track fit. The list of command line options includes:

option	parameter
`-f`	input file name
`-o`	output file name
`-E`	Energy of the muon $[\mathrm{GeV}]$
`-R`	random background rate $[\mathrm{Hz}]$
`-D`	zenith and azimuth angle of the orientation of the PMT

The input file name should be a string containing the wild-card character '%'. This character is replaced by the values of the function type listed in table 3. The resulting string should correspond to a file that contains the binary data of the corresponding PDF. The output file is in ROOT format [9]. The two orientation angles of the PMT should be surrounded by quotes to ensure they are parsed together. The output file contains a set of histograms of the PDF as a function of the arrival time of light of the first hit for different values of the distance of closest approach of the muon track to the PMT.

References

[1] ANTARES-SOFT-2010-002, M. de Jong.

[2] Numerical Recipes in C++, W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Cambridge University Press.

[3] ANTARES-SOFT-2011-XXX, M. de Jong.

[4] Particle Data Group, Phys. Rev. **D66** (2002).

[5] R. Mirani, "Parametrisation of EM-showers in the ANTARES detector volume.", Doctoral thesis in computational physics, University of Amsterdam.

value	method
1	`getDirectLightFromMuon(...)`
2	`getScatteredLightFromMuon(...)`
3	`getDirectLightFromEMshowers(...)`
4	`getScatteredLightFromEMshowers(...)`
5	`getDirectLightFromDeltaRays(...)`
6	`getScatteredLightFromDeltaRays(...)`
13	`getDirectLightFromEMshower(...)`
14	`getScatteredLightFromEMshower(...)`

Table 3: Possible values of the function type and the corresponding member method of the `JPDF` class.

[6] Proceedings of ICRC 2001, "Precise parametrizations of muon energy losses in water", S. Klimushin, E. Bugaev and I. Sokalski.

[7] C. Kopper, "Performance Studies for the KM3NeT Neutrino Telescope.", PhD thesis, University of Erlangen.

[8] David J.L. Bailey, "Monte Carlo tools and analysis methods for understanding the ANTARES experiment and predicting its sensitivity to Dark Matter", PhD thesis, University of Oxford, United Kingdom, 2002.

[9] `http://root.cern.ch/root/`