

The Jpp - JQuadrature package

M. de Jong

July 1, 2019

Abstract

The Jpp - JQuadrature package provides a set of C⁺⁺ methods for the numerical computation of specific integrals.

1 Introduction

This note describes the C⁺⁺ methods that are part of the Jpp framework for the numerical computation of specific integrals.

2 Numerical integration

In general, the integration of a function can be computed numerically by a weighed sum of the function values at some predetermined abscissas, i.e:

$$\int_{x_1}^{x_2} W(x) f(x) dx \simeq \sum_{i=0}^{N-1} w_i f(x_i) \quad (1)$$

where x_1 (x_2) refers to the lower (upper) limit of the integral and $W(x)$ to some weight function. The summation is performed at fixed values, x_i , with weights, w_i . The goal is then to find the values of x_i and w_i that yield the most accurate result for a limited number of points, N . There is extensive literature on the determination of the optimal values for the abscissas and weights [1]. In the following, the classes that are available within the Jpp - JTools package for the numerical computation of function integrals are briefly described.

The base class for the various implementations is a collection of elements, based on the JCollection and JElement2D classes that are part of the Jpp - JTools framework.

```
class JQuadrature :  
    public JCollection<JElement2D_t>  
{}
```

The following classes extend the JCollection class. The methods getX() and getY() of the JElement2D class correspond to the abscissas and weights, respectively.

3 Implementations

3.1 Gauss-Legendre integration

The `JGaussLegendre` class can be used to evaluate the abscissas and weights for $W(x) = 1$.

```
class JGaussLegendre :
    public JQuadrature
{
    JGaussLegendre(const int n, const double eps);
};
```

The first argument of the constructor refers to the number of points and the second (optional) argument to the precision of the evaluation of the abscissa values.

3.2 Gauss-Laguerre integration

The `JGaussLaguerre` class can be used to evaluate the abscissas and weights for $W(x) = x^\alpha e^{-x}$.

```
class JGaussLaguerre :
    JQuadrature
{
    JGaussLaguerre(const int n, const double alf, const double eps);
};
```

The first argument of the constructor refers to the number of points, the second argument to the power, α , and the third (optional) argument to the precision of the evaluation of the abscissa values.

3.3 Gauss-Hermite integration

The `JGaussHermite` class can be used to evaluate the abscissas and weights for $W(x) = e^{-x^2}$.

```
class JGaussHermite :
    public JQuadrature
{
    JGaussHermite(const int n, const double eps);
};
```

The first argument of the constructor refers to the number of points and the second (optional) argument to the precision of the evaluation of the abscissa values.

4 Example

As an example, the effect of the transition time spread (TTS) of the photo-multiplier tube (PMT) on the probability density function (PDF) of the arrival time of light can approximately be determined by the numerical convolution of the PDF and a Gaussian function with a $\sigma = \text{TTS}$.

```
JGaussHermite engine(25);

double x = ..;
double y = 0.0;

for (JGaussHermite::const_iterator j = engine.begin(); j != engine.end(); ++j) {

    const double u = j->getX();
    const double w = j->getY() / sqrt(PI);

    y += w * f(x + u*TTS);
}
```

where x refers to some arrival time and $f()$ to the PDF as a function of the arrival time. The result is stored in y .

References

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