SCINET MATH

NUMERICAL INTEGRATION

Whitepaper Series

1

1. Introduction

Numerical Integration involves the approximate computation of a definite integral using numerical methods. It provides quick and accurate solutions to difficult integrals or integrals without analytical solutions.

Scinet Math implements a broad family of numerical techniques to support the numerical integration. The following numerical integration techniques are provided:

1. Trapezoidal Rule
2. Romberg Integration
3. Simpson’s Rule
4. Gauss-Kronrod Quadrature
5. Gauss-Laguerre Quadrature
6. Gauss-Legendre Quadrature
7. Gauss-Hermite Quadrature
8. Gauss-Chebyshev Quadrature

2

2. Numerical Integration Methods

2.1 Newton-Cotes Integration Methods

The Newton-Cotes formulas are an extremely useful and straightforward family of Numerical Integration techniques. Newton-Cotes integration methods, such as the **Trapezoidal Rule** and the **Simpson's Rule** are supported with the NIntegration class. The class resides in the `Scinet.Math.Integration` namespace and supports the more advanced **Romberg Method** as well. The integration is computed with the **Integrate** method. The following overloaded versions of this method are provided:

```
double Integrate(Function f, double a, double b)
double Integrate(Function f, double a, double b, IntegrationMethod method)
double Integrate(Function f, double a, double b, IntegrationMethod method,
                                        double tolerance, int maxIteration)
```

where

```
Function f                 // the one-variable function to integrate
double a                   // the lower limit of the integral
double b                   // the upper limit of the integral
IntegrationMethod method   // the integration method to use
double tolerance           // The tolerance factor
int maxIteration           // maximum number of iterations
```

The first overload uses the Romberg Method as the default integration method, and the values of 100 and 0.01 for the maximum number of iterations and the tolerance factor, respectively.

The second overload uses the values of 100 and 0.01 for the maximum number of iterations and the tolerance factor, respectively. It allows you to specify which integration method to use.

The third overload allows you to specify the integration method, the tolerance factor and the number of maximum iterations to perform until the computation converges to a value for the specified function in [a, b] interval. It provides greater flexibility.

Trapezoidal Rule

The Trapezoidal Rule connects data points with straight lines and forms a trapezoidal area between the successive data points. The integral of the function is the sum of the areas defined by these trapezoids.

Romberg Integration

Romberg Integration method successively applies the Trapezoidal Rule to estimate the integral of a function more effectively.

Simpson's Rule

Simpson's Integration method uses parabolic arcs to approximate the integral. The data interval is divided into an even number of equal areas with width dx. Scinet Math implements the Simpson's 1/3 rule.

The following code segment shows the NIntegration class in action:

```
double a = 0; // the lower limit of the integral
double b = 3; // the upper limit of the integral
Console.WriteLine("Integral = {0}", NIntegration.Integrate(f, a, b));
Console.WriteLine("Integral = {0}", NIntegration.Integrate(f, a, b,
                                             IntegrationMethod.Simpson));
Console.WriteLine("Integral = {0}", NIntegration.Integrate(f, a, b,
                                 IntegrationMethod.Trapezoid, 0.001, 1000));
```

where f is:

```
double f(double x)
{
      return x * x - 2;
}
```

The output is:

```
Integral = 3        (Romberg Method)
Integral = 3        (Simpson Rule)
Integral = 4.125    (Trapezoidal Rule)
```

Notice that the Trapezoidal rule is a crude method to compute the numerical integration and therefore produces a result with less accuracy.

The Romberg method successively applies the Trapezoidal rule to get a better estimate.

The Simpson's rule provides the exact integral if the input data is parabolic as in this case.

2.2 The Gaussian Quadrature Methods

The Gaussian Quadrature methods employ a different way of calculating integrals with higher accuracy. Instead of sampling the function values at regular intervals of the independent variable, the Gaussian Quadrature zoom in the critical function values. The integral is computed based on the product of the critical function value and special weights.

Scinet Math supports the following Gaussian Quadrature methods:

- Gauss-Kronrod Quadrature
- Gauss-Laguerre Quadrature
- Gauss-Legendre Quadrature
- Gauss-Hermite Quadrature
- Gauss-Chebyshev Quadrature

Gauss-Kronrod Quadrature

The Gauss-Kronrod Quadrature allows the computation of definite integrals in the form of:

$$\int_a^b f(x)\,dx.$$

These types of integrals can be computed approximately by the n-point Gaussian Quadrature:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^n w_i f(x_i).$$

where w_i are the weights and x_i are the abscissas at where the function f is evaluated.

Scinet.Math provides the GaussKronrodQuadrature class to support the method. The class uses the **61-point Gauss-Kronrod Quadrature**.

Example:

Let's compute the following integral with Scinet Math:

$$\int_{-1}^{+1} x^2 - 2 \; dx$$

We define the function f as:

```
double f(double x)
{
      return x * x - 2;
}
```

and use the GaussKronrodQuadrature to compute the integral.

```
GaussKronrodQuadrature quadrature = new GaussKronrodQuadrature ();
double  integral = quadrature.Integrate(f, -1, 1, ref error);
Console.WriteLine("Integral = {0}, Error = {1}", integral, error);
```

The output is:

```
Integral = -3.33333333333333, Error = 8.88178419700125E-16
```

Gauss-Laguerre Quadrature

The Gauss-Laguerre Quadrature evaluates the integrals of the following kind:

$$\int_0^{+\infty} e^{-x} f(x)\, dx.$$

The method calculates the approximate value of the integration by summing the product of Quadrature weights and the values of function f(x) at critical points of x. This can be formulated as:

$$\int_0^{+\infty} e^{-x} f(x)\, dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

Scinet.Math provides the GaussLaguerreQuadrature class to support this method.

Example:
Let's compute the following integral with Scinet Math:

$$\int_0^{\infty} e^{-x} \sin(2x)\, dx$$

When compared to the above formula, it can be seen that the function f(x) is sin(2x). We define the function f as:

```
double f(double x)
{
       return System.Math.Sin(2 * x);
}
```

and use the GaussLaguerreQuadrature to compute the integral.

```
GaussLaguerreQuadrature quadrature = new GaussLaguerreQuadrature ();
Console.WriteLine("Integral = {0}", quadrature.Integrate(f));
```

The output is:

```
Integral = 0.444354044223563
```

Gauss-Legendre Quadrature

The Gauss-Legendre Quadrature uses the Legendre polynomials to obtain the weights and critical points to evaluate the integrals in the [-1, 1] interval.

The method calculates the approximate value of the integration by summing the product of Quadrature weights and the values of function f(x) at critical points of x. This can be formulated as:

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i).$$

Scinet.Math provides the `GaussLegendreQuadrature` class to support this method. The class internally handles the change of interval allowing users to specify any interval of their choosing.

Example:

Let's compute the following integral with Scinet Math:

$$\int_{0}^{5} \sin(2x)\,dx$$

We define the function f as:

```
double f(double x)
{
	return System.Math.Sin(2 * x);
}
```

and use the GaussLegendreQuadrature to compute the integral.

```
GaussLegendreQuadrature quadrature = new GaussLegendreQuadrature();
Console.WriteLine("Integral = {0}", quadrature.Integrate(f, 0, 5));
```

The output is:

```
Integral = 0.919535764538212
```

Gauss-Hermite Quadrature

The Gauss-Hermite Quadrature evaluates the integrals of the following kind:

$$\int_{-\infty}^{+\infty} e^{-x^2} f(x)\,dx.$$

The method calculates the approximate value of the integration by summing the product of Quadrature weights and the values of function f(x) at critical points of x. This can be formulated as:

$$\int_{-\infty}^{+\infty} e^{-x^2} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

Scinet.Math provides the `GaussHermiteQuadrature` class to support this method.

Example:
Let's compute the following integral with Scinet Math:

$$\int_{-\infty}^{+\infty} e^{-x^2} e^{2x}\, dx = \sqrt{\pi}\, e$$

When compared to the above formula, it can be seen that the function f(x) is e^{2x}. We define the function f as:

```
double f(double x)
{
	return System.Math.Exp(2 * x);
}
```

and use the GaussHermiteQuadrature to compute the integral.

```
GaussHermiteQuadrature quadrature = new GaussHermiteQuadrature();
Console.WriteLine("Integral = {0}", quadrature.Integrate(f));
```

The output is:

```
Integral = 4.81775527012573
```

Gauss-Chebyshev Quadrature

The Gauss-Chebyshev Quadrature evaluates the integrals of the following kind:

$$\int_{-1}^{+1} \frac{f(x)}{\sqrt{1-x^2}}\,dx$$

The method calculates the approximate value of the integration by summing the product of Quadrature weights and the values of function f(x) at critical points of x. This can be formulated as:

$$\int_{-1}^{+1} \frac{f(x)}{\sqrt{1-x^2}}\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

where

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right)$$

and

$$w_i = \frac{\pi}{n}.$$

Scinet.Math provides the GaussChebyshevQuadrature class to support this method.
Example:
Let’s compute the following integral with Scinet Math:

$$\int_{-1}^{+1} \frac{1}{\sqrt{1-x^2}}\,dx$$

When compared to the above formula, it can be seen that the function f(x) is 1. We define the function f as:

```
double f(double x)
{
	return 1;
}
```

and use the GaussChebyshevQuadrature to compute the integral.

```
GaussChebyshevQuadrature quadrature = new GaussChebyshevQuadrature ();
Console.WriteLine("Integral = {0}", quadrature.Integrate(f));
```

The output is:

```
Integral = 3.14159265358979
```

3

3 Conclusions

Numerical Integration is an essential component of numerical analysis. **Scinet Math** greatly simplifies the use of Numerical Integration in Microsoft's .NET platform.

A variety of numerical integration techniques are supported. The choice of the method highly depends on the function to be integrated. However, for most of the mathematical functions, Gauss-Kronrod Quadrature, Gauss-Legendre Quadrature and the Romberg method will outperform the other methods of numerical integration in terms of performance and the accuracy. For special functions that are hard to compute the approximate integral, other available methods can be used in computation.

Whitepaper Series
Scinet Math – Numerical Integration

Printed in CANADA, June 2009.

ΘBACS Corporation

Address	5625 Silverdale Dr. N.W., Calgary, Alberta, Canada T3B 4N5
Website	http://www.obacs.com
Information	info@obacs.com
Sales	sales@obacs.com
Technical Support	support@obacs.com