Utility – MathsNET

Special Functions

User Guide

Document Revision 1.0

Trademarks and copyrights

The descriptions and screenshots contained in this document are licensed under the Creative Commons Attribution-ShareAlike (CC-BY-SA) 3.0 license <https://creativecommons.org/licenses/by-sa/3.0/>.

**© Blue Prism Limited, 2001 – 2020**

®“Blue Prism”, the “Blue Prism” logo and Prism device are either trademarks or registered trademarks of Blue Prism Limited and its affiliates. All Rights Reserved

All trademarks are hereby acknowledged and are used to the benefit of their respective owners.  
Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom  
Registered in England: Reg. No. 4260035. Tel: +44 870 879 3000. Web: [www.blueprism.com](file:///C:\Users\adutton\Documents\Rebranding\Templates\www.blueprism.com)

Contents

[1. Introduction 4](#_Toc41212832)

[2. Prerequisites 5](#_Toc41212833)

[3. Configuration 6](#_Toc41212834)

[4. Using the Asset 7](#_Toc41212835)

[4.1 Factorial 7](#_Toc41212836)

[4.2 Logarithmic Factorial Function 7](#_Toc41212837)

[4.3 Binomial Coefficient 8](#_Toc41212838)

[4.4 BinomialLn 8](#_Toc41212839)

[4.5 Multinomial 9](#_Toc41212840)

[4.6 Exponential Integral 9](#_Toc41212841)

[4.7 Gamma Function 10](#_Toc41212842)

[4.8 GammaLn Function 10](#_Toc41212843)

[4.9 Gamma Lower Incomplete (Unregularized) 11](#_Toc41212844)

[4.10 Gamma Upper Incomplete (Unregularized) 11](#_Toc41212845)

[4.11 Gamma Lower Incomplete (Regularized) 12](#_Toc41212846)

[4.12 Gamma Upper Incomplete (Regularized) 12](#_Toc41212847)

[4.13 Gamma Lower Incomplete (Regularized Inverse) 13](#_Toc41212848)

[4.14 DiGamma 13](#_Toc41212849)

[4.15 DiGamma Inverse 14](#_Toc41212850)

[4.16 Euler Beta Function 14](#_Toc41212851)

[4.17 Euler BetaLn Function 15](#_Toc41212852)

[4.18 Incomplete Beta (Unregularized) 15](#_Toc41212853)

[4.19 Beta Regularized 16](#_Toc41212854)

[4.20 Error Function 16](#_Toc41212855)

[4.21 Complimentary Error Function 17](#_Toc41212856)

[4.22 Complimentary Inverse Error Function 17](#_Toc41212857)

[4.23 Logistic Function 18](#_Toc41212858)

[4.24 Logit Function 18](#_Toc41212859)

[4.25 Harmonic 19](#_Toc41212860)

[4.26 Compute the Generalised Harmonic 19](#_Toc41212861)

[4.27 Bessell I0 20](#_Toc41212862)

[4.28 Bessell I1 20](#_Toc41212863)

[4.29 Bessell K 21](#_Toc41212864)

[4.30 Bessell K0 21](#_Toc41212865)

[4.31 Bessell K0e 22](#_Toc41212866)

[4.32 Bessell K1 22](#_Toc41212867)

[4.33 Bessell K1e 23](#_Toc41212868)

[4.34 Struve L0 23](#_Toc41212869)

[4.35 Struve L1 24](#_Toc41212870)

[4.36 Bessel L0 Struve L0 24](#_Toc41212871)

[4.37 Bessel I0M Struve L0 25](#_Toc41212872)

[4.38 Bessel I1M Struve L0 25](#_Toc41212873)

[5. Support 26](#_Toc41212874)

# Introduction

This asset will provide access to the MathNET Numerics Special Functions library. It is part of a series of assets that will offer complimentary mathematical functionality to what is currently possible with Blue Prism alone.

# Prerequisites

To use this asset, you require a licenced Blue Prism installation or trial. More information on Blue Prism can be found here. <https://www.blueprism.com>.

You will also require to download the *MathNET. Numerics dll* from NuGet. The library is found at <https://www.nuget.org/packages/MathNet.Numerics/>

This asset provides 39 mathematical functions. Please see the table of contents for the extensive list.

# Configuration

The asset is downloaded from the Digital Exchange (DX) It is installed int the usual manner using the import item from the file menu.

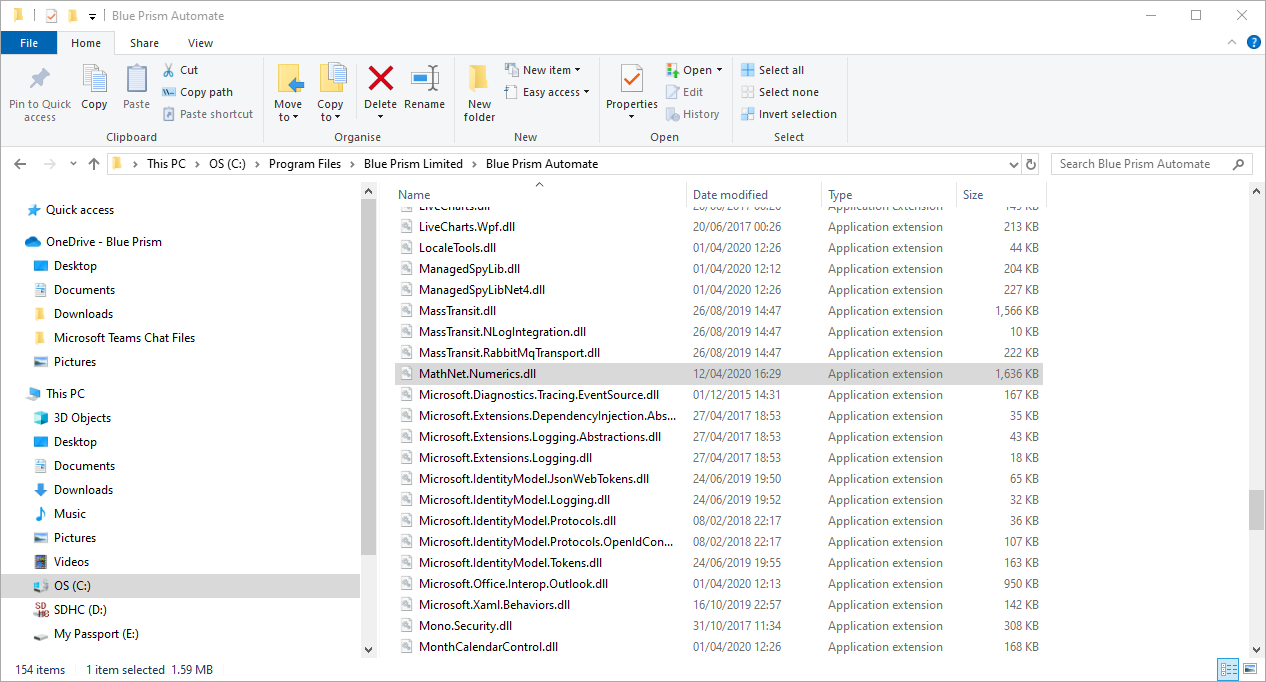
The bprelease file contains:

* Utility MathsNET – Special Functions
* Process Example – MathsNet – Special Functions

You will also be required to download the MathNET. Numerics dll from NuGet. It can be found at <https://www.nuget.org/packages/MathNet.Numerics/>

It needs to be installed in your Blue Prism directory, which will depend on your specific installation. If your Blue Prism is a default install, then the location for its deployment is

C:\program files\Blue Prism Limited\Blue Prism Automate



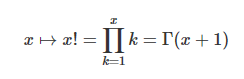
# Using the Asset

This asset is intended to assist anyone requiring extended mathematical functionality beyond the base of what exists in Blue Prism, by default.

## Factorial

The factorial function is a multiplication of all the values up to the xParameter. So if x is 5, the factorial(5) is 5x4x3x2x1 = 120.

The factorial equation

: 

### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | The value of which to determine the factorial. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting factorial value | Number |
| **ResultText** | The resulting factorial value where numerical limits are exceeded. | Text |

## Logarithmic Factorial Function

Computes the logarithmic factorial function. Parameter must be > 0.



### Input:

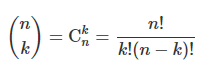
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **ParameterValue** | The input value to calculate the logarithmic factorial. | Text |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting value. | Number |

## Binomial Coefficient

The binomial coefficient is the number of ways of picking k unordered outcomes from n possibilities. This is sometimes known as combinatorics. (source https://en.wikipedia.org/wiki/Binomial\_coefficient)



### Input:

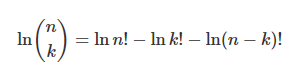
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | The number of elements. | Number |
| **kParameterValue** | The number of unordered elements of the subset. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting coefficient. | Number |

## BinomialLn

Computes the natural logarithm of the binomial coefficient. Ln(n choose k).



### Input:

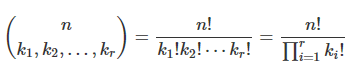
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | The number of elements. | Number |
| **kParameterValue** | The number of unordered elements of the subset. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting natural logarithm of the coefficient. | Number |

## Multinomial

A statistical experiment that consist of n repeated trials. Each trial has a discrete number of possible outcomes. On any given trial, the probability that a particular outcome will occur is constant.



### Input:

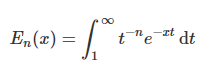
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | The number of elements. | Number |
| **niParameterValue** | The number of unordered elements of the subset. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting coefficient. | Number |

## Exponential Integral

The exponential integral is a special function on the complex plane. It is defined as one particular definite integral of the ratio between an exponential function and its argument.



### Input:

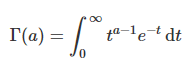
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | The number of elements. | Number |
| **niParameterValue** | The number of unordered elements of the subset. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting coefficient. | Number |

## Gamma Function

The gamma function is an extension of the factorial function to complex numbers. The gamma function is defined for all complex numbers except negative integers.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | The input value to apply to the gamma function. | Text |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## GammaLn Function

The gammaln function will return the natural logarithm of the gamma function, using zParameterValue which must be positive.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | The input value to apply to the gamma function.. | Text |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## Gamma Lower Incomplete (Unregularized)

The lower incomplete gamma function is a special function that is a solution to various mathematical problems such as integrals. The lower incomplete gamma is defined as an integral from zero to a variable upper limit.

Returns the lower incomplete gamma function gamma(a,x) = int(exp(-t)t^(a-1),t=0..x) for real a>0, x>0.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Complex parameter. Real part is positive. | Number |
| **xParameterValue** | Complex parameter. Real part is positive. |  |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## Gamma Upper Incomplete (Unregularized)

The upper incomplete gamma function is a special function that is a solution to various mathematical problems such as integrals. The upper incomplete gamma is defined as an integral from zero to infinity.

Returns the upper incomplete gamma function gamma(a,x) = int(exp(-t)t^(a-1),t=0..x) for real a>0, x>0.



### Input:

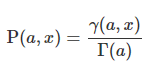
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Complex parameter. Real part is positive. | Number |
| **xParameterValue** | Complex parameter. Real part is positive. |  |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## Gamma Lower Incomplete (Regularized)

Returns the lower incomplete regularized gamma function P(a,x) = 1/Gamma(a) \* int(exp(-t)t^(a-1),t=0..x) for real a>0,x>0.



### Input:

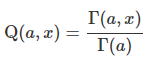
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Complex parameter. Real part is positive. | Number |
| **xParameterValue** | Complex parameter. Real part is positive. |  |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## Gamma Upper Incomplete (Regularized)

The upper incomplete gamma function is a special function that is a solution to various mathematical problems such as integrals. The upper incomplete gamma is defined as an integral from zero to infinity.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Complex parameter. Real part is positive. | Number |
| **xParameterValue** | Complex parameter. Real part is positive. |  |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## Gamma Lower Incomplete (Regularized Inverse)

Returns the inverse P^(-1) of the regularized lower incomplete regularized gamma function P(a,x) = 1/Gamma(a) \* int(exp(-t)t^(a-1),t=0..x) for real a>0,x>0, such that P^(-1)(a,P(a,x)) == x.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Complex parameter. Real part is positive. | Number |
| **xParameterValue** | Complex parameter. Real part is positive. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## DiGamma

Computes the Digamma function which is mathematically defined as the derivative of the logarithm of the gamma function. This implementation is based on Jose Bernardo Algorithms AS 103: Psi (Digamma) Function, Applied Statistics, Volume 25 Number 3, 1976, pages 315-317.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Complex parameter. Real part is positive. | Number |
| **xParameterValue** | Complex parameter. Real part is positive. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Digamma value. | Number |

## DiGamma Inverse

Computes the inverse Digamma function: This is the inverse of the logarithm of the gamma function. This function will only return solutions that are positive. This implementation is based on the bisection method. <https://en.wikipedia.org/wiki/Digamma_function>



### Input:

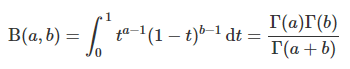
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Integer parameter. Real part is positive. | Number |
| **xParameterValue** | Integer parameter. Real part is positive. | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Digamma value. | Number |

## Euler Beta Function

Computes the Euler Beta function. <https://en.wikipedia.org/wiki/Beta_function>



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | Floating Point Value | Number |
| **wParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting beta value. | Number |

## Euler BetaLn Function

Computes the natural logarithm of the Euler Beta function, evaluated at z,w. <https://en.wikipedia.org/wiki/Beta_function>



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | Floating Point Value | Number |
| **wParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting BetaLn value. | Number |

## Incomplete Beta (Unregularized)

Returns the lower incomplete (unregularized) beta function B(a,b,x) = int(t^(a-1)\*(1-t)^(b-1),t=0...x) for real a>0, b>0, 1>=x>=0.. <https://en.wikipedia.org/wiki/Beta_function#Incomplete_beta_function>

  
Input:

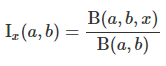
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Floating Point Value. | Number |
| **bParameterValue** | Floating Point Value | Number |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting gamma value. | Number |

## Beta Regularized

Returns the regularized lower incomplete beta function I\_x(a,b) = 1/Beta(a,b) \* int(t^(a-1)\*(1-t)^(b-1),t=0...x) for real a>0, b>0, 1>=x>=0.. <https://en.wikipedia.org/wiki/Beta_function>



### Input:

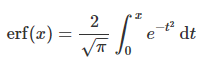
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **aParameterValue** | Floating Point Value | Number |
| **bParameterValue** | Floating Point Value | Number |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting beta value. | Number |

## Error Function

Calculates the Error Function. <https://en.wikipedia.org/wiki/Error_function>



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting error value. | Number |

## Complimentary Error Function

Returns the complimentary error function evaluated at given value z.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting complimentary error function value. | Number |

## Complimentary Inverse Error Function

Calculates the complimentary Inverse Error Function evaluated at z.. <https://en.wikipedia.org/wiki/Error_function>



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **zParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting error value. | Number |

## Logistic Function

Returns the logistic function. https://en.wikipedia.org/wiki/Logistic\_function



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting complimentary error function value. | Number |

## Logit Function

Calculates the Inverse of the logistic function.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **yParameterValue** | For y between 0 and 1 (where the function is real valued). | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting logit value. | Number |

## Harmonic

Computes the t’th harmonic value.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **tParameterValue** | Integer Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting harmonic value. | Number |

## Compute the Generalised Harmonic

Compute the generalized harmonic number of order n of m. (1+1/2^m+1/3m+....1/n^m)



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | Integer Value | Number |
| **mParameterValue** | Floating Point Value |  |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting error value. | Number |

## Bessell I0

Returns the modified Bessel function of first kind, order 0 of the argument. The function is defined as i0(x) = j0(ix). The range is partioned into the two intervals [0,8] and [8,infinity]. Chebyshev polynomial expansions are employed in each interval.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell I0 function value. | Number |

## Bessell I1

Returns the modified Bessel function of first kind, order 1 of the argument. The function is defined as i1(x) = -i j1(ix). The range is partioned into the two intervals [0,8] and [8,infinity]. Chebyshev polynomial expansions are employed in each interval.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell I1 function value. | Number |

## Bessell K

Returns the modified Bessel function of the second kind. BesselK(n,z) is a solution to the modified Bessel differential equation.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | Floating Point Value | Number |
| **zParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessel K function value. | Number |

## Bessell K0

Returns the modified Bessel function of the second kind, order 0 of the argument. The range is partitioned into the two intervals [0,8] and [8, infinity]. Chebyshev polynomial expansions are employed in each interval.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessel I1 function value. | Number |

## Bessell K0e

Returns the modified Bessel function of the second kind. BesselK(n,z) is a solution to the modified Bessel differential equation.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | Floating Point Value | Number |
| **zParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessel K function value. | Number |

## Bessell K1

Returns the modified Bessel function of the second kind order 1 of the argument. The range is partitioned into the two intervals [0,2] and (2,Infinity). Chebyshev polynomial expansions are employed in each interval.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessel I1 function value. | Number |

## Bessell K1e

Returns the modified Bessel function of the second kind.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **nParameterValue** | Floating Point Value | Number |
| **zParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell K function value. | Number |

## Struve L0

Returns the modified Struve function of order 0.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Struve L0 function value. | Number |

## Struve L1

Returns the modified Struve function of order 1.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell K function value. | Number |

## Bessel L0 Struve L0

Returns the difference between the Bessel I0 and Struve L0 functions.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell I1 function value. | Number |

## Bessel I0M Struve L0

Returns the modified Struve function of order 1.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell K function value. | Number |

## Bessel I1M Struve L0

Returns the difference between the Bessel I1 and Struve L1 functions.



### Input:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **xParameterValue** | Floating Point Value | Number |

### Output:

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Data Type** |
| **Result** | The resulting Bessell I1 function value. | Number |

# Support

This asset is provided free-of-charge by Blue Prism. Blue Prism does not provide formal support of this asset. Please direct any questions you have, related to this asset, to the Digital Exchange Community page:

<https://community.blueprism.com/communities/community-home?CommunityKey=1e516cfe-4d1f-4de9-a9eb-58d15bf38c81>