Symbolic Math Toolbox and Extended Symbolic Math Toolbox Release Notes

Note There were no significant updates to the Symbolic Math Toolbox or Extended Symbolic Math Toolbox for Release 14 with Service Pack 1.

The Symbolic Math Toolbox Release Notes also provide information about recent versions of the product, in case you are upgrading from a version prior to Version 3.1.

Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

Contents

1

Symbolic Math Toolbox 3.1 and Extended Symbolic Math 3.1 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Symbolic Math Toolbox 3.1 and the Extended Symbolic Math Toolbox 3.1.

If you are upgrading from a version earlier than 3.0.1 (Release 13 with Service Pack 1), you should also see "New Features" on page 2-2 in the Symbolic Math Toolbox 3.0.1 Release Notes.

The Symbolic Math Toolbox 3.1 and the Extended Symbolic Math Toolbox 3.1 contain the following new features:

Rounding Operations

The following new functions perform rounding operations on symbolic arrays:

- `ceil` — Round a number `x` to the nearest integer greater than or equal to `x`.
- `fix` — Round toward zero.
- `floor` — Round a number `x` to the nearest integer less than or equal to `x`.
- `frac` — Compute the fractional part of a number.
- `round` — Round a number to the nearest integer.

For example,

```
x = sym([2.5; -9.639])
[fix(x) floor(x) round(x) ceil(x) frac(x)]

x =
```

```
       5/2
 -9639/1000
```

```
ans =

[         2,         2,         3,         3,       1/2]
[        -9,       -10,       -10,        -9, -639/1000]
```

Quotient and Remainder for Division of Integers and Polynomials

The new function `quorem` computes the quotient and remainder for division of integers and polynomials. For example,

```
syms x y
p = x^3-2*x+5
[q,r] = quorem(x^5,p)

p =

x^3-2*x+5

q =

x^2+2

r =

-5*x^2-10+4*x
```

Dirac and Step Functions

The following new functions compute the Dirac delta and Heaviside functions:

- `dirac` — Compute the Dirac delta function.
- `heaviside` — Compute the Heaviside step function.

For example,

```
dirac([-1 0 1])

ans =

     0   Inf     0
heaviside([-1 0 1])

ans =

     0   NaN     1
```

Sorting Symbolic Expressions

The new function `sort` sorts symbolic expressions. For example,

```
syms a b c d e x
sort([a c e b d])

ans =

[ a, b, c, d, e]
sort([a c e b d]*x.^(0:4).')

ans =

x^4*d+x^3*b+e*x^2+x*c+a
```

Coefficients of Multivariable Expressions

The new function `coeffs` computes coefficients of a multivariate polynomial. For example,

```
syms c t x y
t = 2 + (3 + 4*log(x))^2 - 5*log(x);
coeffs(expand(t))

ans =

[ 11, 19, 16]
```

```
z = 3*x^2*y^2 + 5*x*y^3
        [c,t] = coeffs(z,y)

z =

3*x^2*y^2+5*x*y^3



c =

[ 3*x^2,   5*x]
t =

[ y^2, y^3]
```

Multidimensional Symbolic Arrays

The new function `reshape` reshapes symbolic arrays. For example,

```
syms x
A = reshape(x.^(1:9),1,3,3)


A(:,:,1) =

[   x, x^2, x^3]


A(:,:,2) =

[ x^4, x^5, x^6]


A(:,:,3) =

[ x^7, x^8, x^9]
```

Conversion to Nondouble Numeric Data Types

The following new functions enable you to convert symbolic arrays to nondouble numeric data types:

- `int8` — Convert a symbolic matrix to signed 8-bit integers.
- `int16` — Convert a symbolic matrix to signed 16-bit integers.
- `int32` — Convert a symbolic matrix to signed 32-bit integers.
- `int64` — Convert a symbolic matrix to signed 64-bit integers.
- `single` — Convert a number to single precision.
- `uint8` — Convert a symbolic matrix to unsigned 8-bit integers.
- `uint16` — Convert a symbolic matrix to unsigned 16-bit integers.
- `uint32` — Convert a symbolic matrix to unsigned 32-bit integers.
- `uint64` — Convert a symbolic matrix to unsigned 64-bit integers.

Logarithms to Base 2 and Base 10

The following new functions enable you to compute the logarithm of symbolic arrays to base 2 and base 10:

- `log10` — Compute base 10 logarithm.
- `log2` — Compute base 2 logarithm.

Modulus After Division

The new function `mod` computes modulus after division. For example,

```
syms x
mod(x^3-2*x+999,10)
       x^3+8*x+9

ans =

x^3+8*x+9
```

2

Symbolic Math Toolbox 3.0.1 and Extended Symbolic Math 3.0.1 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Symbolic Math Toolbox 3.0.1 and the Extended Symbolic Math Toolbox 3.0.1.

If you are upgrading from a release earlier than Release 12 (MATLAB 6.0), then you should also see “Major Bug Fixes” on page 3-3 in the Symbolic Math Toolbox 2.1.2 Release Notes.

Maple Version 8

The Symbolic Math Toolboxes now use the Maple version 8 kernel to perform calculations. Maple is mathematical software developed by Waterloo Maple, Inc.

For a complete list of the new features in Maple 8, see

http://www.maplesoft.com/products/Maple8/whatsnew/features.shtml

Support for the Macintosh Platform

The Symbolic Math Toolboxes are now supported on the Macintosh platform.

Improved Memory Management

The Symbolic Math Toolboxes now have improved memory management, which reduces memory usage and memory addressing errors on large symbolic computations.

3

Symbolic Math Toolbox and Extended Symbolic Math Toolbox 2.1.2 Release

Notes

Major Bug Fixes

The Symbolic Math Toolbox 2.1.2 includes several bug fixes. The particularly important bug fixes are described in the online documentation.