

Symbolic Math Toolbox™

Release Notes

How to Contact MathWorks



www.mathworks.com Web
comp.soft-sys.matlab Newsgroup
www.mathworks.com/contact_TS.html Technical Support



suggest@mathworks.com Product enhancement suggestions
bugs@mathworks.com Bug reports
doc@mathworks.com Documentation error reports
service@mathworks.com Order status, license renewals, passcodes
info@mathworks.com Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Symbolic Math Toolbox™ Release Notes

© COPYRIGHT 2004–2011 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Summary by Version	1
Version 5.7 (R2011b) Symbolic Math Toolbox Software	5
Version 5.6 (R2011a) Symbolic Math Toolbox Software	10
Version 5.5 (R2010b) Symbolic Math Toolbox Software	14
Version 5.4 (R2010a) Symbolic Math Toolbox Software	19
Version 5.3 (R2009b) Symbolic Math Toolbox Software	25
Version 5.2 (R2009a) Symbolic Math Toolbox Software	29
Version 5.1 (R2008b) Symbolic Math Toolbox Software	34
Version 5.0 (R2008a+) Symbolic Math Toolbox Software	35
Version 4.9 (R2007b+) Symbolic Math Toolbox Software	36
Version 3.2.3 (R2008a) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software	44
Version 3.2.2 (R2007b) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software	45

Version 3.2 (R2007a) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software	46
Version 3.1.5 (R2006b) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software	47
Version 3.1 (R14) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software	48
Compatibility Summary for Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software	53

Summary by Version

This table provides quick access to what's new in each version. For clarification, see “Using Release Notes” on page 3.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Latest Version V5.7 (R2011b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.6 (R2011a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.5 (R2010b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.4 (R2010a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.3 (R2009b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.2 (R2009a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V5.1 (R2008b)	No	No Note If you are upgrading from a version before 4.9, see the release notes for “Version 4.9 (R2007b+) Symbolic Math Toolbox Software” on page 36.	Bug Reports Includes fixes

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
V5.0 (R2008a+)	No	No Note If you are upgrading from a version before 4.9, see the release notes for “Version 4.9 (R2007b+) Symbolic Math Toolbox Software” on page 36.	Bug Reports Includes fixes
V4.9 (R2007b+)	Yes Details	Yes Summary	Bug Reports Includes fixes
V3.2.3 (R2008a)	No	No	Bug Reports Includes fixes
V3.2.2 (R2007b)	No	No	Bug Reports Includes fixes
V3.2 (R2007a)	Yes Details	No	Bug Reports Includes fixes
V3.1.5 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V3.1.4 (R2006a)	No	No	Bug Reports Includes fixes
V3.1.3 (R14SP3)	No	No	No bug fixes
V3.1.2 (R14SP2)	No	No	Bug Reports Includes fixes
V3.1.1 (R14SP1)	No	No	No bug fixes
V3.1 (R14)	Yes Details	No	No bug fixes

Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks® products required for this product (for example, MATLAB® or Simulink®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What Is in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product release appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time

and as more information becomes available. Bug Reports include provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Documentation on the MathWorks Web Site

Related documentation is available on mathworks.com for the latest release and for previous releases:

- Latest product documentation
- Archived documentation

Version 5.7 (R2011b) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.7 (R2011b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are:

- “MATLAB Editor Now Supports MuPAD Program Files” on page 6
- “dsolve, expand, int, simple, simplify, and solve Accept More Options” on page 6
- “New read Function Reads MuPAD Program Files in MATLAB” on page 6
- “New symprod Function Computes Products of Series” on page 7
- “New hessian Function Computes Hessian Matrices” on page 7
- “New gradient Function Computes Vector Gradients” on page 7
- “New erfc Function Computes the Complementary Error Function” on page 7
- “New psi Function Computes the Digamma and Polygamma Functions” on page 7
- “New wrightOmega Function Computes the Wright omega Function” on page 7
- “New simplifyFraction Function Simplifies Expressions” on page 7
- “New MuPAD simplifyRadical Function Simplifies Radicals in Arithmetical Expressions” on page 8
- “pretty Function Now Uses Abbreviations in Long Output Expressions for Better Readability” on page 8

- “MuPAD normal Function Accepts the New Expand Option” on page 8
- “Modified MuPAD groebner Library” on page 8
- “MuPAD groebner::gbasis Function Accepts the New Factor and IgnoreSpecialCases Options” on page 9
- “New MuPAD Functions for Computing Logarithms” on page 9
- “Functionality Being Removed or Changed” on page 9

MATLAB Editor Now Supports MuPAD Program Files

You can open and edit MuPAD® program files (.mu files) in the MATLAB Editor. MATLAB Editor supports syntax highlighting and smart indenting for these files.

dsolve, expand, int, simple, simplify, and solve Accept More Options

`dsolve` now accepts the `IgnoreAnalyticConstraints` and `MaxDegree` options.

`expand` now accepts the `ArithmeticOnly` and `IgnoreAnalyticConstraints` options.

`int` now accepts the `IgnoreAnalyticConstraints`, `IgnoreSpecialCases`, and `PrincipalValue` options.

`simple` now accepts the `IgnoreAnalyticConstraints` option.

`simplify` now accepts the `IgnoreAnalyticConstraints`, `Seconds`, and `Steps` options.

`solve` now accepts the `IgnoreAnalyticConstraints`, `IgnoreProperties`, `MaxDegree`, `PrincipalValue`, and `Real` options.

New read Function Reads MuPAD Program Files in MATLAB

`read` simplifies using your own MuPAD procedures in MATLAB. See “Before Calling a Procedure” for details.

New symprod Function Computes Products of Series

symprod computes definite and indefinite products of symbolic series.

New hessian Function Computes Hessian Matrices

hessian computes the Hessian matrix of a scalar function.

New gradient Function Computes Vector Gradients

gradient computes the vector gradient of a scalar function in Cartesian coordinates. In MuPAD, the new `linalg::gradient` function is equivalent to `linalg::grad`.

New erfc Function Computes the Complementary Error Function

erfc computes the complementary error function.

New psi Function Computes the Digamma and Polygamma Functions

psi computes the digamma and polygamma functions.

New wrightOmega Function Computes the Wright omega Function

wrightOmega computes the Wright omega function.

New simplifyFraction Function Simplifies Expressions

simplifyFraction returns a simplified form of a fraction where both numerator and denominator are polynomials and their greatest common divisor is 1. In MuPAD, the new `simplifyFraction` function is equivalent to `normal`.

New MuPAD `simplifyRadical` Function Simplifies Radicals in Arithmetical Expressions

The new MuPAD `simplifyRadical` function is equivalent to the MuPAD `radsimp` function.

`pretty` Function Now Uses Abbreviations in Long Output Expressions for Better Readability

`pretty` uses abbreviations when presenting symbolic results in the MATLAB Command Window. This new format of presenting symbolic results enhances readability of long output expressions.

MuPAD `normal` Function Accepts the New `Expand` Option

The MuPAD `normal` function accepts the new `Expand` option that determines whether numerators and denominators of fractions are expanded.

Compatibility Considerations

In previous releases, `normal` returned a fraction with the expanded numerator and denominator by default. Now the default setting is that `normal` can return factored expressions in numerators and denominators. In explicit calls to `normal`, you can use the `Expand` option to get the same behavior as in previous releases.

If a function calls `normal` internally, then that function can return its results in a different form. These new results are mathematically equivalent to the results that you get in previous releases. Many MuPAD library functions can call `normal`.

Modified MuPAD `groebner` Library

All functions of the MuPAD `groebner` library now can accept and return polynomials with arbitrary arithmetical expressions.

MuPAD groebner::gbasis Function Accepts the New Factor and IgnoreSpecialCases Options

With the Factor option, groebner::gbasis returns a set of lists, such that:

- Each list is the Groebner basis of an ideal.
- The union of these ideals is a superset of the ideal given as input, and a subset of the radical of that ideal.

With the IgnoreSpecialCases option, groebner::gbasis handles all coefficients in all intermediate results as nonzero unless these coefficients are equal to 0 for all parameter values.

New MuPAD Functions for Computing Logarithms

The new MuPAD log2 and log10 functions compute logarithms to the bases 2 and 10, respectively. Also, in MuPAD log(x) is now an alias for ln(x).

Functionality Being Removed or Changed

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
emlBlock	Warns	matlabFunctionBlock	Replace all instances of emlBlock with matlabFunctionBlock.
Real and IgnoreProperties options in MuPAD ode::solve	Warns	IgnoreSpecialCases or IgnoreAnalyticConstraints	Try using IgnoreSpecialCases or IgnoreAnalyticConstraints instead.

Version 5.6 (R2011a) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.6 (R2011a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are:

- “Expression Wrapping of Math Output in the MuPAD Notebook Interface” on page 11
- “Symbolic Solver Handles More Non-Algebraic Equations” on page 11
- “Improved Performance in the Ordinary Differential Equation Solver” on page 11
- “Improved Performance for Polynomial Arithmetic Operations” on page 11
- “New MuPAD polylib::subresultant Function Computes Subresultants of Polynomials” on page 11
- “MuPAD partfrac Function Accepts the New List Option” on page 11
- “New MuPAD inverf and inverfc Functions Compute the Inverses of Error Functions” on page 12
- “New MuPAD numlib::checkPrimalityCertificate Function Tests Primality Certificates” on page 12
- “New Demos” on page 12
- “Functionality Being Removed or Changed” on page 12

Expression Wrapping of Math Output in the MuPAD Notebook Interface

The new default format of presenting results enhances readability by wrapping long output expressions, including long numbers, fractions and matrices.

Symbolic Solver Handles More Non-Algebraic Equations

The enhanced `rationalize` function in MuPAD helps the symbolic solver to handle more systems of non-algebraic equations. In particular, this improvement enables the toolbox to solve more systems of trigonometric equations.

Improved Performance in the Ordinary Differential Equation Solver

The ordinary differential equation solver demonstrates better performance.

Improved Performance for Polynomial Arithmetic Operations

The MuPAD functions `gcdex`, `partfrac`, `polylib::resultant`, and `solveLib::pdioe` now demonstrate better performance.

New MuPAD `polylib::subresultant` Function Computes Subresultants of Polynomials

`polylib::subresultant` computes subresultants of two polynomials or polynomial expressions.

MuPAD `partfrac` Function Accepts the New List Option

With the new `List` option, `partfrac` returns a list consisting of the numerators and denominators of the partial fraction decomposition.

New MuPAD `inverf` and `inverfc` Functions Compute the Inverses of Error Functions

The `inverf` function computes the inverse of the error function.

The `inverfc` function computes the inverse of the complementary error function.

New MuPAD `numlib::checkPrimalityCertificate` Function Tests Primality Certificates

`numlib::checkPrimalityCertificate` tests primality certificates returned by `numlib::proveprime`. For information about proving primality of numbers, see “Proving Primality” in the MuPAD documentation.

New Demos

There are three new demos that show how to solve equations and compute derivatives and integrals:

- Solving Algebraic and Differential Equations
- Differentiation
- Integration

To run the new demos, enter `symeqndemo`, `syndiffdemo`, or `symintdemo` in the MATLAB Command Window.

Functionality Being Removed or Changed

Functionality	What Happens When You	Use This Instead	Compatibility Considerations
---------------	-----------------------	------------------	------------------------------

	Use This Functionality?		
MuPAD <code>matchlib::analyze</code>	Errors	MuPAD <code>prog::exprtree</code>	To visualize expressions, use <code>prog::exprtree</code> .
MuPAD <code>prog::testcall</code>	Errors	Nothing	No replacement
MuPAD <code>prog::testerrors</code>	Errors	Nothing	No replacement
Old syntax of MuPAD <code>prog::getOptions</code>	Errors	The new syntax	Update all instances of <code>prog::getOptions</code> calls using the new syntax.
Old syntax of MuPAD <code>prog::trace</code>	Errors	The new syntax	Update all instances of <code>prog::trace</code> calls using the new syntax.

Version 5.5 (R2010b) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.5 (R2010b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are:

- “sym Function Creates Matrices of Symbolic Variables” on page 15
- “generate::Simscape Function Generates Simscape Equations from MuPAD Expressions” on page 15
- “MuPAD Code Generation Functions Accept the New NoWarning Option” on page 15
- “Improved MuPAD Hyperlink Dialog Box” on page 15
- “MuPAD Notebook Highlights Matched and Unmatched Delimiters” on page 15
- “Improved Performance When Solving Linear Systems in a Matrix Form” on page 16
- “MuPAD Solver for Ordinary Differential Equations Handles More Equation Types” on page 16
- “New Syntax for the MuPAD prog::getOptions Function” on page 16
- “New Syntax for the MuPAD prog::trace Function” on page 16
- “Improved Interface for Arithmetical Operations on Polynomials” on page 16
- “MuPAD igcd Function Now Accepts Complex Numbers as Arguments” on page 17
- “Enhanced Solver For Factorable Polynomial Systems” on page 17

- “MuPAD Now Evaluates Large Sums with Subtractions Faster” on page 17
- “MuPAD freeIndets Function Accepts the New All Option” on page 17
- “Functionality Being Removed or Changed” on page 18

sym Function Creates Matrices of Symbolic Variables

The `sym` function now provides a shortcut for creating vectors and matrices of symbolic variables.

For more information, see “Creating a Matrix of Symbolic Variables”.

generate::Simscape Function Generates Simscape Equations from MuPAD Expressions

The new MuPAD function `generate::Simscape` converts MuPAD expressions to Simscape™ equations.

MuPAD Code Generation Functions Accept the New NoWarning Option

MuPAD functions `generate::C`, `generate::fortran`, `generate::MATLAB`, and `generate::Simscape` accept the new `NoWarning` option. The option suppresses all warnings issued by these functions.

Improved MuPAD Hyperlink Dialog Box

Creating and editing links in MuPAD has become easier with the improved Hyperlink dialog box.

MuPAD Notebook Highlights Matched and Unmatched Delimiters

MuPAD Notebook now can notify you about matched and unmatched delimiters such as parentheses, brackets, and braces.

Improved Performance When Solving Linear Systems in a Matrix Form

MuPAD `linalg::matlinsolve` function, which solves linear systems of equations in a matrix form, demonstrates better performance.

MuPAD Solver for Ordinary Differential Equations Handles More Equation Types

Enhanced MuPAD solver handles more first-order nonlinear and third-order linear ordinary differential equations. The solver demonstrates improved performance.

New Syntax for the MuPAD `prog::getOptions` Function

The `prog::getOptions` function that collects and verifies options within a procedure has the new syntax.

Compatibility Considerations

The new syntax is not valid in MuPAD versions earlier than 5.5. The old syntax is supported in MuPAD 5.5, but will be removed in a future release.

New Syntax for the MuPAD `prog::trace` Function

The `prog::trace` function used for debugging has the new syntax. The function observes entering and exiting the MuPAD functions.

Compatibility Considerations

The new syntax is not valid in MuPAD versions earlier than 5.5. The old syntax is not supported in MuPAD 5.5.

Improved Interface for Arithmetical Operations on Polynomials

Improved interface for arithmetical operations between polynomials and arithmetical expressions. In previous releases, to perform an arithmetical operation on a polynomial and an arithmetical expression, you must explicitly

convert that expression to a polynomial of the corresponding type. Now, when you operate on a polynomial and an arithmetical expression, MuPAD internally converts the arithmetical expression to a polynomial and performs the calculation.

MuPAD igcd Function Now Accepts Complex Numbers as Arguments

The MuPAD `igcd` function, which computes the greatest common divisor of integers, now accepts complex numbers. Both real and imaginary parts of accepted complex numbers must be integers or arithmetic expressions that represent integers.

Enhanced Solver For Factorable Polynomial Systems

The MuPAD `solve` function performs better on factorable polynomial systems.

MuPAD Now Evaluates Large Sums with Subtractions Faster

MuPAD performs evaluations of large sums that contain subtractions faster than in previous releases.

Compatibility Considerations

In MuPAD, the difference operator (`-`) no longer invokes the `_subtract` function. Instead, it invokes the `_plus` and `_negate` functions. For example, `a - b` is equivalent to `_plus(a, _negate(b))`.

MuPAD freeIndets Function Accepts the New All Option

The `freeIndets` function accepts the new `All` option. With this option, `freeIndets` does not exclude the 0th operand from the list of free identifiers.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
diff and int methods for inputs of the char type	Warns	sym	Use the sym method instead.
MuPAD matchlib::analyze	Warns	MuPAD prog::exptree	To visualize expressions, use prog::exptree.
MuPAD prog::testcall	Warns	None	No replacement
MuPAD prog::testerrors	Warns	None	No replacement
The following options in MuPAD prog::trace: <ul style="list-style-type: none"> • All • Backup • Force • Name • Proc • Plain • Width 	Errors	None	No replacement. These options are not supported in the current release.
Global properties in MuPAD	Errors	Assumptions on each variable	Make assumptions on each variable instead.

Version 5.4 (R2010a) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.4 (R2010a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are:

- “When Opening Notebook, MuPAD Can Jump to Particular Locations” on page 20
- “simscapeEquation Function Generates Simscape Equations from Symbolic Expressions” on page 20
- “New Calling Syntax for the sort Function” on page 20
- “Changes in the symengine Function” on page 21
- “64-Bit GUI Support for Macintosh” on page 21
- “New MuPAD Print Preview Dialog” on page 21
- “Improved Configure MuPAD Dialog Box” on page 21
- “MuPAD Support for Basic Arithmetic Operations for Lists” on page 21
- “Improved Performance When Operating on Matrices with Symbolic Elements” on page 21
- “Enhanced MuPAD divide Function” on page 21
- “Improved Performance for Operations on Polynomials” on page 22
- “MuPAD coeff Function Accepts the New All Option” on page 22
- “MuPAD expand Function Accepts the New ArithmeticOnly Option” on page 22

- “MuPAD expand Function Now Expands Powers of Products” on page 22
- “New Calling Syntax for MuPAD rationalize Function” on page 23
- “Enhanced MuPAD simplify and Simplify Functions” on page 23
- “MuPAD subs Function Accepts the New EvalChanges Option” on page 23
- “MuPAD Solver for Ordinary Differential Equations Handles More Equation Types” on page 23
- “The digits and vpa Functions: Compatibility Considerations” on page 23
- “Functionality Being Removed or Changed” on page 24

When Opening Notebook, MuPAD Can Jump to Particular Locations

The `mupad` command that opens a MuPAD notebook now supports references to particular places inside a notebook. You can create a link target inside a notebook and refer to it when opening a notebook.

`simscapeEquation` Function Generates Simscape Equations from Symbolic Expressions

The new `simscapeEquation` command represents symbolic expressions in the form of Simscape equations. For more information, see [Generating Simscape Equations in the Symbolic Math Toolbox™ documentation](#).

New Calling Syntax for the `sort` Function

The `sort` function that sorts the element of symbolic arrays and polynomials has the new syntax and set of options.

Compatibility Considerations

In previous releases, the `sort` function flattened symbolic matrices to vectors before sorting the elements. Now the `sort` function sorts the elements of each column or each row of a symbolic matrix. If you want to obtain the same results as in the previous release, flatten the symbolic matrix before sorting it: `sort(A(:))`.

Changes in the symengine Function

The toolbox no longer supports the ability to choose an alternative symbolic engine.

64-Bit GUI Support for Macintosh

MuPAD now supports 64-bit graphical user interfaces (such as notebooks and Editor and Debugger windows) for a 64-bit Macintosh® operating system.

New MuPAD Print Preview Dialog

Adjusting MuPAD documents for printing is easier with the new Print Preview dialog. You can view one or several pages, zoom in and out, switch between page orientations, adjust the page settings without closing the dialog, and print the page or save it to PDF format.

Improved Configure MuPAD Dialog Box

Specifying the default settings for graphical user interfaces, such as notebooks and Editor and Debugger windows, has become easier with the improved configuration dialog box.

MuPAD Support for Basic Arithmetic Operations for Lists

Basic arithmetic operations now work for lists.

Improved Performance When Operating on Matrices with Symbolic Elements

MuPAD demonstrates better performance when handling some linear algebra operations on matrices containing symbolic elements.

Enhanced MuPAD divide Function

Enhanced MuPAD `divide` function computes the quotient and remainder for division of multivariate polynomials.

Improved Performance for Operations on Polynomials

Improved performance for conversions involving polynomials. Improved performance for operations on polynomials including evaluation, multiplication, and division.

Compatibility Considerations

If the coefficients of a polynomial contain the variables of the polynomial itself, the form of results returned by the MuPAD `poly` function can differ from previous releases. In previous releases, the `poly` function converted such coefficients to monomials. Now the `poly` function can return the coefficients of the original expression as coefficients in the resulting polynomial. To get the same behavior as in previous releases, use `expr` to convert an original polynomial into an expression, and then call the `poly` function. For example, the following call exercises the old behavior: `poly(expr(p), [y, x])`.

MuPAD `coeff` Function Accepts the New `All` Option

The `coeff` function accepts the `newAll` option. With this option, `coeff` returns all coefficients of a polynomial including those equal to 0.

MuPAD `expand` Function Accepts the New `ArithmeticOnly` Option

The `expand` function accepts the new `ArithmeticOnly` option. The option allows you to expand a sum without expanding trigonometric expressions and special functions in its terms. Technically, the option omits overloading the `expand` function for each term of the original expression.

MuPAD `expand` Function Now Expands Powers of Products

The `expand` function now expands powers of products such as $(xy)^n$ for positive x and y . When called with the `IgnoreAnalyticConstraints` option, the function expands the power of products for arbitrary terms.

New Calling Syntax for MuPAD rationalize Function

The `rationalize` function that transforms an arbitrary expression into a rational expression has the new syntax and set of options.

Compatibility Considerations

The new syntax is not valid in MuPAD versions earlier than 5.4. The old syntax is supported in MuPAD 5.4, but will be removed in a future release.

Enhanced MuPAD `simplify` and `Simplify` Functions

Enhanced simplification functions, `simplify` and `Simplify`, demonstrate better results for expressions involving trigonometric and hyperbolic functions, square roots, and sums over roots of unity.

MuPAD `subs` Function Accepts the New `EvalChanges` Option

The `subs` function now accepts the new `EvalChanges` option. By default, `subs` does not evaluate an expression after making substitutions. With this option, `subs` evaluates all subexpressions that contain substitutions.

MuPAD Solver for Ordinary Differential Equations Handles More Equation Types

Enhanced MuPAD solver handles more second-order linear and first-order nonlinear ordinary differential equations. The solver demonstrates improved performance.

The `digits` and `vpa` Functions: Compatibility Considerations

It is no longer possible to set the number of digits to 1 when using the `digits` and `vpa` functions. The Symbolic Math Toolbox software version number 4.9 and lower allowed you to set the number of digits to 1.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
MuPAD Domain <code>Dom::Ideal</code>	Errors	<code>groebner</code>	Represent ideals as lists, and use functions of the <code>groebner</code> package instead.
MuPAD student library	Errors	<code>plot::Integral</code> and <code>linalg</code>	Use <code>plot::Integral</code> and the <code>linalg</code> package instead.
MuPAD relation option in <code>simplify</code>	Errors	None	No replacement
Global property	Warns	Assumptions on each variable	Make assumptions on each variable instead.

Version 5.3 (R2009b) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.3 (R2009b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are described here:

- “Support for Windows x64 and 64-Bit Macintosh” on page 25
- “sym and syms Use Reserved Words as Variable Names” on page 26
- “Toolbox Now Displays Floating-Point Results with Their Original Precision” on page 26
- “New MuPAD Preference Pref::outputDigits Controls Floating-Point Outputs” on page 27
- “Solver for Ordinary Differential Equations Handles More Equation Types” on page 27
- “MuPAD limit Function Supports Limits for Incomplete Gamma Function and Exponential Integral Function” on page 27
- “Enhanced Simplification Routines for MuPAD Special Functions” on page 27
- “Enhanced MuPAD combine Function for Logarithms” on page 27
- “MuPAD normal Function Accepts New Options” on page 27
- “Functionality Being Removed or Changed” on page 27

Support for Windows x64 and 64-Bit Macintosh

The toolbox now supports 64-bit Windows® and Macintosh operating systems. If you work in the MuPAD Notebook Interface on a 64-bit Macintosh operating

system, MuPAD runs a 64-bit engine with 32-bit graphical user interfaces, such as notebooks and Editor and Debugger windows.

sym and syms Use Reserved Words as Variable Names

`sym` and `syms` commands now treat reserved MuPAD words, except `pi`, as variable names.

Compatibility Considerations

In previous releases, the reserved words returned MuPAD values. If your code uses the reserved words as MuPAD commands, modify your code and use the `evalin` command with the reserved word as a name. For example, use `evalin(symengine, 'beta')`.

Toolbox Now Displays Floating-Point Results with Their Original Precision

The toolbox now displays the floating-point results with the original precision with which the toolbox returned them.

Compatibility Considerations

In previous releases, the toolbox displayed floating-point results with the current precision. You must update the existing code that relies on the output precision for displaying floating-point numbers. Use `digits` to set the precision you need before computing such results. The toolbox displays the results with the same number of digits it used to compute the results. The toolbox also can increase the specified precision of calculations by several digits.

In previous releases, `sym(A, 'f')` represented numbers in the form $(2^e + N \cdot 2^{(e - 52)})$ or $-(2^e + N \cdot 2^{(e - 52)})$, with integers for N and e , and $N \neq 0$. Now `sym(A, 'f')` displays results in the rational form that actually represents the double-precision floating-point numbers.

New MuPAD Preference `Pref::outputDigits` Controls Floating-Point Outputs

New preference `Pref::outputDigits` controls the precision MuPAD uses to display floating-point results.

Solver for Ordinary Differential Equations Handles More Equation Types

Enhanced solvers handle more equation types of second-order homogeneous linear ordinary differential equations. The solver demonstrates improved performance.

MuPAD limit Function Supports Limits for Incomplete Gamma Function and Exponential Integral Function

Enhanced limit function now can compute limits for incomplete Gamma function and exponential integral function.

Enhanced Simplification Routines for MuPAD Special Functions

Enhanced simplification routines for MuPAD `hypergeom`, `mejerG`, and `bessel` special functions.

Enhanced MuPAD `combine` Function for Logarithms

Enhanced `combine` function demonstrates better performance for logarithms.

MuPAD `normal` Function Accepts New Options

The `normal` command now accepts the options `NoGcd`, `ToCancel`, `Rationalize`, `Recursive`, and `Iterations`. The options control costly operations, such as recognizing greatest common divisors and algebraic dependencies.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
MuPAD Domain <code>Dom::Ideal</code>	Warns	<code>groebner</code>	Represent ideals as lists, and use functions of the <code>groebner</code> package instead.
MuPAD student library	Warns	<code>plot::Integral</code> and <code>linalg</code>	Use <code>plot::Integral</code> and the <code>linalg</code> package instead.
<code>d in char(A, d)</code>	Warns	None	No replacement
MuPAD relation option in <code>simplify</code>	Warns	None	No replacement

Version 5.2 (R2009a) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.2 (R2009a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are described here:

- “dsolve Accepts the New Option IgnoreAnalyticConstraints” on page 30
- “emlBlock Function Generates Embedded MATLAB Function Blocks from Symbolic Objects” on page 30
- “matlabFunction Improves Control over Input and Output Parameters” on page 30
- “Enhancements to Object-Oriented Programming Capabilities” on page 31
- “generate::MATLAB Function Converts MuPAD Expressions to MATLAB Code” on page 31
- “MuPAD IgnoreAnalyticConstraints Option Specifies That Core Functions Apply Common Algebraic Assumptions to Simplify Results” on page 31
- “MuPAD Outputs Contain Abbreviations for Better Readability” on page 32
- “MuPAD Solver for Ordinary Differential Equations Handles More Equation Types” on page 32
- “MuPAD limit Function Now Can Compute Limits for Piecewise Functions” on page 32
- “New and Improved MuPAD Special Functions” on page 32
- “New Calling Syntax for Test Report Function prog::tcov” on page 32
- “New Demos” on page 33

dsolve Accepts the New Option IgnoreAnalyticConstraints

The `dsolve` command now accepts the option `IgnoreAnalyticConstraints`. The option controls the level of mathematical rigor that the solver uses on the analytical constraints on the solution. By default, the solver ignores all analytical constraints.

Compatibility Considerations

The results of the `dsolve` command can differ from those returned in the previous release. If you want to obtain the same solutions as in the previous release, set the value of the option `IgnoreAnalyticConstraints` to `none`.

emlBlock Function Generates Embedded MATLAB Function Blocks from Symbolic Objects

The new `emlBlock` command converts symbolic expressions to Embedded MATLAB® Function Blocks. You can use these blocks in any Simulink installation, even those without a Symbolic Math Toolbox license. For more information, see [Generating Embedded MATLAB Blocks in the Symbolic Math Toolbox documentation](#).

matlabFunction Improves Control over Input and Output Parameters

`matlabFunction` now accepts multiple expressions and cell arrays of symbolic arrays as input parameters. The function now allows you to specify the names of the output parameters.

Compatibility Considerations

In previous releases, the default name of an output variable was `RESULT`. Now the default names of the output variables coincide with the names you use to call `matlabFunction`. You must update existing code that relies on the default output name `RESULT`. You can change your code using any of these methods:

- Define the name of an output variable as `RESULT`.
- Change the name of an input variable to `RESULT`.

- Throughout your code change the variable name from RESULT to the input name.

Enhancements to Object-Oriented Programming Capabilities

The Symbolic Math Toolbox product uses some object-oriented programming features to implement symbolic objects. Major enhancements to object-oriented programming capabilities enable easier development and maintenance of large applications and data structures. For a full description of object-oriented features, see the MATLAB Object-Oriented Programming documentation.

Compatibility Considerations

It is no longer possible to add methods to @sym by creating a @sym directory containing custom methods.

For an empty x , `sym(x)` returns a symbolic object of the same size as x . In previous releases, `sym(x)` returned a symbolic object of size 0-by-0 for an empty x .

generate::MATLAB Function Converts MuPAD Expressions to MATLAB Code

The new `generate::MATLAB` command converts MuPAD expressions, equations, and matrices to MATLAB formatted strings.

MuPAD IgnoreAnalyticConstraints Option Specifies That Core Functions Apply Common Algebraic Assumptions to Simplify Results

The new `IgnoreAnalyticConstraints` option allows the use of a set of simplified mathematical rules when solving equations, simplifying expressions, or integrating. For example, this option applies practical, but not generally correct rules for combining logarithms: $\ln(a) + \ln(b) = \ln(a \cdot b)$

MuPAD Outputs Contain Abbreviations for Better Readability

The new default format of presenting results enhances readability of long output expressions by using abbreviations.

MuPAD Solver for Ordinary Differential Equations Handles More Equation Types

The solver now can handle more than 200 additional types of second-order ordinary differential equations. The solver demonstrates improved performance.

MuPAD limit Function Now Can Compute Limits for Piecewise Functions

The enhanced `limit` function computes limits of piecewise functions including bidirectional and one-sided limits.

New and Improved MuPAD Special Functions

MuPAD includes the following new special functions:

- `laguerreL` represents Laguerre's L function.
- `erfc(x,n)` returns iterated integrals of the complementary error function.
- `meijerG` represents the Meijer G function.

The `hypergeom` special function demonstrates better performance.

New Calling Syntax for Test Report Function `prog::tcov`

The `prog::tcov` function that inspects the data collected during the code execution has the new syntax and set of options.

Compatibility Considerations

The new syntax is not valid in MuPAD versions earlier than 5.2. MuPAD 5.2 does not support the earlier syntax.

New Demos

To see new demos that use MuPAD Notebook Interface, type `mupadDemo` at the MATLAB command line or click MuPAD Notebooks Demo.

Version 5.1 (R2008b) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.1 (R2008b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Bug Reports Includes fixes

There are no new features or changes in this version.

Note If you are upgrading from a version before 4.9, see the release notes for “Version 4.9 (R2007b+) Symbolic Math Toolbox Software” on page 36.

Version 5.0 (R2008a+) Symbolic Math Toolbox Software

This table summarizes what's new in Version 5.0 (R2008a+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Bug Reports Includes fixes

There are no new features or changes in this version.

Note If you are upgrading from a version before 4.9, see the release notes for “Version 4.9 (R2007b+) Symbolic Math Toolbox Software” on page 36.

Version 4.9 (R2007b+) Symbolic Math Toolbox Software

This table summarizes what's new in Version 4.9 (R2007b+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are described here:

- “MuPAD Engine Replaces Maple Engine” on page 36
- “New MuPAD Language and Libraries Supplant Extended Symbolic Math Toolbox Software” on page 41
- “New MuPAD Help Viewer (GUI)” on page 41
- “New MuPAD Notebook Interface (GUI)” on page 41
- “New MuPAD Editor and Debugger (GUI)” on page 42
- “New Functionality for Communication Between MATLAB Workspace and MuPAD” on page 42
- “New symengine Command for Choosing a Maple Engine” on page 43
- “New matlabFunction Generates MATLAB Functions” on page 43

MuPAD Engine Replaces Maple Engine

The default Symbolic Math Toolbox engine is now the MuPAD engine. For more information, see the “MuPAD in Symbolic Math Toolbox” chapter in the Symbolic Math Toolbox User’s Guide.

Compatibility Considerations

The new engine causes many computed results to differ from those returned by previous versions of Symbolic Math Toolbox software.

General Differences.

- Many computations return in a permuted order (such as $a + b$ instead of $b + a$).
- Some computations return in a different, mathematically equivalent form (such as $(\cos(x))^2$ instead of $1 - (\sin(x))^2$).
- `diff(dirac(t))` returns `dirac(t,1)` instead of `dirac(1,t)`.
- `sym(x, 'f')` no longer produces strings of the form `hex digits*2^n`. Instead the strings have the form $(2^e + N \cdot 2^{e-52})$, where N and e are integers.
- For toolbox calculations, some symbols can only be used as symbolic variables, and not in strings: `E`, `I`, `D`, `O`, `beta`, `zeta`, `theta`, `psi`, `gamma`, `Ci`, `Si`, and `Ei`. This is because those symbols represent MuPAD reserved words, and are interpreted as the MuPAD word if you pass them as strings. The words `Ci`, `Si`, `Ei` represent special mathematical functions: the cosine integral, sine integral, and exponential integral respectively.
- Error and warning message IDs may have changed.
- Performance of numerical integration is slower than in previous versions.
- Subexpressions, calculated by the `subexpr` function, may be different than in previous versions.
- The `pretty` function no longer uses partial subexpressions (with syntax `%n`).

Calculus.

- `Int` no longer evaluates some integrals, including many involving Bessel functions.
- `symsum(sin(k*pi)/k,0,n)` no longer evaluates to `pi`.

Linear Algebra.

- The output of `colspace` may differ from previous versions, but it is mathematically equivalent.
- The `eig` function may return eigenvalues in a different order than previous versions. Expressions returned by `eig` may be larger than in previous versions.

- The `jordan` function may return diagonal subblocks in a different order than previous versions.
- `svd` may return singular values in a different order than previous versions.

Simplification.

- The `coeffs` function may return multivariable terms in a different order than in previous versions.
- The `expand` function may return some trig and exponential expressions differently than in previous versions.
- The `simplify` function involving radicals and powers make fewer assumptions on unknown symbols than in previous versions.
- The `subexpr` function may choose a different subexpression to be the common subexpression than in previous versions.
- Subexpressions no longer have partial subexpressions (previous syntax `%n`).
- The `solve` function returns solutions with higher multiplicity only when solving a single polynomial.
- $\text{acot}(-x) = -\text{acot}(x)$ instead of $\pi - \text{acot}(x)$ as in previous versions.
- $\text{acoth}(-x) = -\text{acoth}(x)$ instead of $2*\text{acoth}(0) - \text{acoth}(x)$ as in previous versions.
- The `simple` function has several differences:
 - The 'how' value `combine(trig)` has been replaced with `combine(sincos)`, `combine(sinhcosh)`, and `combine(ln)`.
 - The 'how' values involving `convert` have been replaced with `rewrite`.
 - A new 'how' value of `mlsimplify(100)` indicates the MuPAD function `Simplify(...,Steps=100)` simplified the expression.
 - Simplifications such as $(\sin(x)^2)^{(1/2)}$ to $\sin(x)$ are no longer performed, since the MuPAD language is careful not to make assumptions about the sign of $\sin(x)$.

Conversion.

- Arithmetic involving the `vpa` function uses the current number of digits of precision. Variable precision arithmetic may have different rounding behaviors, and answers may differ in trailing digits (trailing zeros are now suppressed).
- The `char` function returns strings using MuPAD syntax instead of Maple™ syntax.
- Testing equality does not compare strings as in previous versions; the symbolic engine equality test is used.
- Saving and loading symbolic expressions is compatible with previous versions, except when the symbolic contents use syntax or functions that differ between Maple or MuPAD engines. For example, suppose you save the symbolic object `sym('transform::fourier(f,x,w)')`, which has MuPAD syntax. You get a MATLAB error if you try to open the object while using a Maple engine.
- LaTeX output from the `latex` function may look different than before.
- C and Fortran code generated with the `ccode` and `fortran` functions may be different than before. In particular, generated files have intermediate expressions as “optimized” code. For more information, see the “Generating C or Fortran Code” section of the User’s Guide.
- pretty output may look different than before.

Equation Solving.

- `solve` returns solutions with higher multiplicity only when solving a single polynomial.
- `solve` may return a different number of solutions than before.
- Some calls to `dsolve` that used to return results involving `lambertw` now return no solution.
- `dsolve` can now use the variable `C`.
- Some `dsolve` results are more complete (more cases are returned).
- Some `dsolve` results are less complete (not all previous answers are found).

- `finverse` may be able to find inverses for different classes of functions than before.
- When `finverse` fails to find an explicit inverse, it produces different output than before.

Transforms.

- Fourier and inverse Fourier transforms return the MuPAD form `transform::fourier` when they cannot be evaluated. For example,

```
h = sin(x)/exp(x^2);
FF = fourier(h)

FF =
transform::fourier(sin(x)/exp(x^2), x, -w)
```

The reason for this behavior is the MuPAD definition of Fourier transform and inverse Fourier transform differ from their Symbolic Math Toolbox counterparts by the sign in the exponent:

	Symbolic Math Toolbox definition	MuPAD definition
Fourier transform	$F(w) = \int_{-\infty}^{\infty} f(x)e^{-iwx} dx$	$F(w) = \int_{-\infty}^{\infty} f(x)e^{iwx} dx$
Inverse Fourier transform	$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w)e^{iwx} dw$	$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w)e^{-iwx} dw$

- Several Fourier transforms can no longer be calculated, especially those involving Bessel functions.
- `ztrans` and `iztrans` may return more complicated expressions than before.

Special Mathematical Functions.

- The three-parameter Riemann Zeta function is no longer supported.

- `heaviside(0) = 0.5`; in previous versions it was undefined.

maple.

- The `maple`, `mhelp`, and `procread` functions error, unless a Maple engine is installed and selected with `symengine`.

New MuPAD Language and Libraries Supplant Extended Symbolic Math Toolbox Software

The functionality of the MuPAD language, together with the included libraries, goes far beyond that of the previous Symbolic Math Toolbox software. However, it is not identical to that of the previous Extended Symbolic Math Toolbox™ software. The differences between these software packages are beyond the scope of these release notes.

You can access the MuPAD language in several ways:

- To learn the commands, syntax, and functionality of the language, use the MuPAD Help browser, or read the Tutorial.
- Use a MuPAD notebook, which contains an integrated help system for the language syntax.
- Use the new `evalin` function or `feval` function to access the MuPAD language at the MATLAB command line. More detail is available in the “Calling Built-In MuPAD Functions from the MATLAB Command Window” section of the User’s Guide.

New MuPAD Help Viewer (GUI)

The MuPAD help viewer contains complete documentation of the MuPAD language, and of the MuPAD Notebook Interface. For more information, see the “Getting Help for MuPAD” section of the User’s Guide.

New MuPAD Notebook Interface (GUI)

A MuPAD notebook is an interface for performing symbolic math computations with embedded math notation, graphics, animations, and text. It also enables you to share, document, and publish your calculations and graphics. For example, the MuPAD help viewer is essentially a special MuPAD notebook.

For more information, see the “Calculating in a MuPAD Notebook” section of the User’s Guide.

New MuPAD Editor and Debugger (GUI)

The MuPAD Editor GUI enables you to write custom symbolic functions and libraries in the MuPAD language. The Debugger enables you to test your code. For more information, consult the MuPAD help viewer.

New Functionality for Communication Between MATLAB Workspace and MuPAD

Function	Use
<code>doc(symengine,...)</code>	Access the MuPAD Help browser.
<code>evalin(symengine,...)</code>	Use MuPAD functionality in the MATLAB workspace.
<code>feval(symengine,...)</code>	Use MuPAD functionality in the MATLAB workspace.
<code>getVar</code>	Copy expressions residing in a MuPAD notebook into the MATLAB workspace.
<code>mupad</code>	Launch a MuPAD notebook .
<code>mupadwelcome</code>	Access MuPAD GUIs .
<code>reset(symengine,...)</code>	Clear the MuPAD engine for the MATLAB workspace.
<code>setVar</code>	Copy expressions residing in the MATLAB workspace into a MuPAD notebook.
<code>symvar</code>	Produce a list of symbolic objects in an expression.

For more information, see the “Integration of MuPAD and MATLAB” section of the User’s Guide.

New symengine Command for Choosing a Maple Engine

If you own a compatible version of a Maple software, you can choose to have Symbolic Math Toolbox software use the Maple engine instead of a MuPAD engine. You might want to do this if you have existing Maple programs. Choose the engine by entering `symengine` at the MATLAB command line; this brings up a GUI for making your choice.

New matlabFunction Generates MATLAB Functions

The new `matlabFunction` generates MATLAB functions from symbolic expressions. `matlabFunction` writes the generated code to a file or creates a function handle. You can use the generated function handles and files in any MATLAB installation, even those without a Symbolic Math Toolbox license. For more information, see “Generating MATLAB Functions” in the User’s Guide.

Version 3.2.3 (R2008a) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software

This table summarizes what's new in Version 3.2.3 (R2008a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Bug Reports Includes fixes

There are no new features or changes in this version.

Version 3.2.2 (R2007b) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software

This table summarizes what's new in Version 3.2.2 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Bug Reports Includes fixes

There are no new features or changes in this version.

Version 3.2 (R2007a) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software

This table summarizes what's new in Version 3.2 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features and changes introduced in this version are described here:

Maple10 Access Added for Linux 64-bit Processors and Intel Macintosh Platforms

MATLAB now supports Maple Version 10 on 32-bit Windows, 32- and 64-bit Linux® platforms, and the Intel® and PowerPC® Macintosh platforms.

Version 3.1.5 (R2006b) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software

This table summarizes what's new in version 3.1.5 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are described here:

Change in call to code generation package using the maple function

Calling a function in code generation package using Maple software now requires you to explicitly include the package name. For example,

```
maple('codegen[fortran](x^2-4)');
```

The generated code output using these methods is unaffected by this change.

Compatibility Considerations

In previous versions, functions in the code generation package of Maple software were made automatically available using the Maple with command, and did not require the package name. For example

```
maple('fortran(x^2-4)');
```

This sometimes caused a conflict when assigning to Maple variables having the same name as a function in the code generation package.

Version 3.1 (R14) Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software

This table summarizes what's new in version 3.1 (R14):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	No

New features and changes introduced in this version are described here:

- “Rounding Operations” on page 48
- “Quotient and Remainder for Division of Integers and Polynomials” on page 49
- “Dirac and Step Functions” on page 49
- “Sorting Symbolic Expressions” on page 50
- “Coefficients of Multivariable Expressions” on page 50
- “Multidimensional Symbolic Arrays” on page 51
- “Conversion to Nondouble Numeric Data Types” on page 51
- “Logarithms to Base 2 and Base 10” on page 52
- “Modulus After Division” on page 52

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)

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


Compatibility Summary for Symbolic Math Toolbox and Extended Symbolic Math Toolbox Software

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided with the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V5.7 (R2011b)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • “MuPAD normal Function Accepts the New Expand Option” on page 8 See “Functionality Being Removed or Changed” on page 9.
V5.6 (R2011a)	See “Functionality Being Removed or Changed” on page 12.
V5.5 (R2010b)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • “New Syntax for the MuPAD prog::getOptions Function” on page 16 • “New Syntax for the MuPAD prog::trace Function” on page 16 • “MuPAD Now Evaluates Large Sums with Subtractions Faster” on page 17

Version (Release)	New Features and Changes with Version Compatibility Impact
	See “Functionality Being Removed or Changed” on page 18.
V5.4 (R2010a)	<p>See the Compatibility Considerations subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> • “New Calling Syntax for the sort Function” on page 20 • “Improved Performance for Operations on Polynomials” on page 22 • “New Calling Syntax for MuPAD rationalize Function” on page 23 • “The digits and vpa Functions: Compatibility Considerations” on page 23 <p>See “Functionality Being Removed or Changed” on page 24.</p>
V5.3 (R2009b)	<p>See the Compatibility Considerations subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> • “sym and syms Use Reserved Words as Variable Names” on page 26 • “Toolbox Now Displays Floating-Point Results with Their Original Precision” on page 26

Version (Release)	New Features and Changes with Version Compatibility Impact
	See “Functionality Being Removed or Changed” on page 27.
V5.2 (R2009a)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • “dsolve Accepts the New Option IgnoreAnalyticConstraints” on page 30 • “matlabFunction Improves Control over Input and Output Parameters” on page 30 • “Enhancements to Object-Oriented Programming Capabilities” on page 31 • “New Calling Syntax for Test Report Function prog::tcov” on page 32
V5.1 (R2008b)	None
V5.0 (R2008a+)	None
V4.9 (R2007b+)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • “MuPAD Engine Replaces Maple Engine” on page 36
V3.2.3 (R2008a)	None
V3.2.2 (R2007b)	None
V3.2 (R2007a)	None

Version (Release)	New Features and Changes with Version Compatibility Impact
V3.1.5 (R2006b)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none">• “Change in call to code generation package using the maple function” on page 47
V3.1.4 (R2006a)	None
V3.1.3 (R14SP3)	None
V3.1.1 (R14SP1)	None
V3.1 (R14)	None