Wavelet Toolbox™ Release Notes
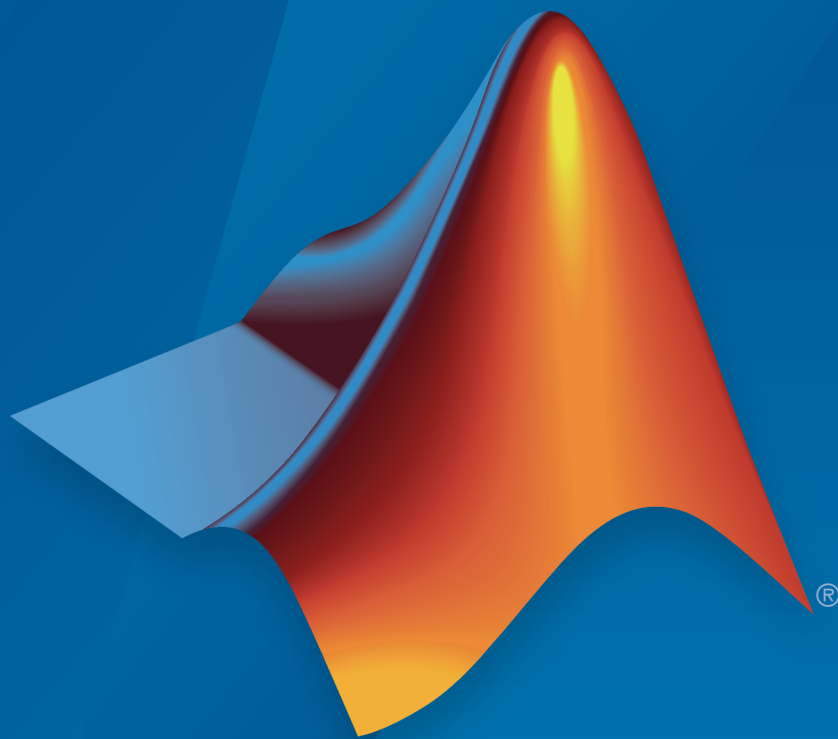
MATLAB®

How to Contact MathWorks

	Latest news:	www.mathworks.com
	Sales and services:	www.mathworks.com/sales_and_services
	User community:	www.mathworks.com/matlabcentral
	Technical support:	www.mathworks.com/support/contact_us
	Phone:	508-647-7000

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

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.

Contents

R2015a

R2014b

Bug Fixes

R2014a

R2013b

R2013a

Bug Fixes

R2012b

Bug Fixes

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

Bug Fixes

R2009a

R2008b

R2008a

R14SP3

No New Features or Changes

R14SP2

No New Features or Changes

R2016a

Version: 4.16

New Features

Bug Fixes

Compatibility Considerations

Maximal Overlap Discrete Wavelet Packet Transform: Perform nondecimated wavelet packet analysis on arbitrary-length signals

This release adds support for the maximal overlap discrete wavelet packet transform (MODWPT) for 1-D signals. You can decompose signals using `modwpt` and invert the transform using `imodwpt`. Also, you can obtain MODWPT details using `modwptdetails`. For an example of using wavelet packets, see "Wavelet Packets: Decomposing the Details".

Wavelet Synchrosqueezing: Sharpen time-frequency estimates and extract signal modes

This release adds support for the wavelet synchrosqueezed transform and mode extraction for 1-D signals. Wavelet synchrosqueezing is a time-frequency reassignment technique that enables you to reconstruct the signal from the reassigned transform. This technique enables you to extract and visualize oscillatory modes in the signal. To obtain the synchrosqueezed transform of a signal, use `wsst`. To invert the transform, use `iwsst`. You can determine or extract time-frequency ridges in the synchrosqueezed transform with `wsstridge`. For an example of synchrosqueezing, see "Time-Frequency Reassignment and Mode Extraction with Synchrosqueezing".

Wavelet Coherence: Compare time-varying frequency content between signals

This release adds the `wcoherence` function, which computes the magnitude-squared wavelet coherence of two input signals. The `wcoherence` function also computes the wavelet cross spectrum. Wavelet coherence is useful for detecting common time-localized oscillations in nonstationary, bivariate signals. `wcoherence` also provides visualizations that show the magnitude-squared coherence, cross-spectrum phase, and the cone of influence. The phase plot is helpful in determining the lead-lag relationships between the signals. The cone of influence demonstrates where edge effects become significant. For an example of using `wcoherence`, see "Compare Time-Frequency Content in Signals with Wavelet Coherence".

Compatibility Considerations

`wcoher` is not recommended. Update code that uses `wcoher` to use `wcoherence` instead.

Fejer-Korovkin filter with 18 coefficients

This release adds support for Fejer-Korovkin (`fejerkorovkin`) scaling and wavelet filters with 18 coefficients. The valid short name is `'fk18'`.

Maximum derivative output of `gauswavf` and `cgauwavf`

As of R2016a, the highest order derivative supported for the Gaussian (`gauswavf`) and complex Gaussian wavelet (`cgauwavf`) is 8.

Compatibility Considerations

Specifying a derivative order greater than 8 produces an error. In code that uses `gauswavf` or `cgauwavf`, update these functions to use a derivative value from 1 to 8. The requirement to have Symbolic Math Toolbox™ has been removed.

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
`wcoher`	Still runs	`wcoherence`	Replace all instances of `wcoher` with `wcoherence`.
`gauswavf` and `cgauwavf`	Errors when the order of the derivative is greater than 8		Update instances of `gauswavf` and `cgauwavf` to use a maximum derivative value of 8.

R2015b

Version: 4.15

New Features

Bug Fixes

Maximal Overlap Discrete Wavelet Transform: Perform nondecimated analysis on arbitrary-length signals and obtain multiscale variance and correlation estimates

This release adds support for the maximal overlap discrete wavelet transform (MODWT) for 1-D signals. You can decompose signals using modwt and invert the transform using imodwt. Additionally, you can obtain a MODWT-based multiresolution analysis using modwtmra. You can also obtain wavelet variance, correlation, and cross-correlation sequence estimates with confidence intervals using modwtvar, modwtcorr, and modwtxcorr.

Frequency-Localized Wavelets: Perform more frequency-localized discrete and continuous wavelet analysis

This release adds new frequency-localized wavelets for continuous and discrete wavelet analysis. The bump wavelet is a frequency-localized wavelet with an adjustable center frequency and bandwidth.

You can use the bump wavelet with cwtft. For discrete decimated and nondecimated wavelet or wavelet packet analysis, use the new Fejer-Korovkin family of frequency-localized orthogonal wavelets. To obtain information on the Fejer-Korovkin wavelets, enter `waveinfo('fk')` at the MATLAB® command prompt. To obtain the Fejer-Korovkin filters, use wfilters or fejerkorovkin. You can specify the Fejer-Korovkin filters in all discrete wavelet and wavelet packet command line and interactive applications using the short name, `'fk'` with a valid filter number. For example, `wavedec(data,N,'fk8')` or `modwt(data,'fk8')`.

Time-Frequency Analysis: Convert scale to frequency to interpret the continuous wavelet transform as a time-frequency transform

This release adds scale-to-frequency conversion for cwtft and cwt. `cwtft` returns the scale-to-frequency conversions as a field in the structure array output. `cwt` accepts an optional sampling interval input, which enables you to output scale-to-frequency conversions. This release also adds a featured example, Time-Frequency Analysis with the Continuous Wavelet Transform.

Application examples: Analyze financial data and physiologic signals using wavelets

This release introduces new examples for analyzing financial data and physiologic signals using wavelets. The financial example, Wavelet Analysis of Financial Data, shows how you can use wavelets to analyze multiscale volatility in financial time series data and explore multiscale correlation in bivariate time series data. The physiologic signal analysis example, Wavelet Analysis of Physiologic Signals, showcases QRS detection in the electrocardiogram using wavelets, wavelet coherence, and time-frequency analysis.

R2015a

Version: 4.14.1

Bug Fixes

Compatibility Considerations

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
ndwt, indwt, ndwt2, indwt2	Errors	swt, iswt, swt2, iswt2	Replace all instances of ndwt, indwt, ndwt2, and indwt2 with the corresponding function for the stationary wavelet transform.

R2014b

Version: 4.14

Bug Fixes

R2014a

Version: 4.13

New Features

Bug Fixes

Dual-Tree Wavelet Transforms

This release introduces a new example, Dual-Tree Wavelet Transforms, which demonstrates the advantages of the dual-tree discrete wavelet transform (DWT) over the critically sampled DWT. The example illustrates the approximate shift invariance and directional selectivity of the complex dual-tree wavelet transform. These properties enable the dual-tree wavelet transform to outperform the critically sampled DWT in a number of applications.

R2013b

Version: 4.12

New Features

Bug Fixes

Two-Dimensional Continuous Wavelet Transform (2-D CWT)

This release introduces the 2-D continuous wavelet transform (CWT) for images. The 2-D CWT provides information about images at specified scales, rotation angles, and positions in the plane. Applications of the 2-D CWT include:

- Fault detection in images
- Object recognition
- Fringe pattern profilometry

For information on how to implement the 2-D CWT at the MATLAB command line, see cwtft2.

To use `cwtft2` in the Wavelet Toolbox™ interactive tool, enter

```
>> wavemenu
```
Then, from the **Two-dimensional** tools section, select **Continuous Wavelet Transform 2-D**. See 2-D Continuous Wavelet Transform App for more information on the 2-D CWT app.

`cwtft2` supports both isotropic and anisotropic 2-D wavelets. Use isotropic wavelets to perform pointwise analysis in images or when oriented features are not relevant. Use anisotropic wavelets when your goal is to detect oriented features.

`cwtft2` implements the 2-D CWT using the 2-D discrete Fourier transform. Use cwtftinfo2 to obtain the 2-D Fourier transforms of the supported analyzing wavelets.

Dual-Tree Transforms and Double-Density Transforms

This release introduces two types of 1-D and 2-D oversampled (frame) wavelet perfect reconstruction filter banks. For 1-D wavelet analysis, use dddtree to obtain the following wavelet transforms:

- Complex dual-tree
- Double-density
- Complex dual-tree double-density

For 2-D wavelet analysis, use dddtree2 to obtain the following wavelet transforms:

- Double-density

- Real oriented dual-tree
- Complex oriented dual-tree
- Real oriented double-density dual-tree
- Complex oriented double-density dual-tree

The dual-tree and double-density transforms mitigate a number of shortcomings of the critically sampled discrete wavelet transform. The double-density and dual-tree transforms achieve directional selectivity and approximate shift invariance with significantly less computational cost than the undecimated discrete wavelet transform.

R2013a

Version: 4.11

Bug Fixes

R2012b

Version: 4.10

Bug Fixes

R2012a

Version: 4.9

New Features

Matching Pursuit

In R2012a you can decompose a 1-D signal in a dictionary of time/frequency or time/scale atoms with matching pursuit.

Representing a signal in a union of time-frequency/time-scale bases can provide sparser signal representations than attainable with any single basis. Matching pursuit uses iterative greedy algorithms to reduce the computational complexity of searching through a redundant dictionary.

Wavelet Toolbox software supports basic matching pursuit, orthogonal matching pursuit, and weak orthogonal matching pursuit at the command line with wmpdictionary and wmpalg. You can also perform matching pursuit with the interactive wavemenu tool.

You can build dictionaries using several internally supported options or provide your own custom dictionaries. See Matching Pursuit for background information and examples.

R2011b

Version: 4.8

New Features

Bug Fixes

Compatibility Considerations

Fourier Transform Based Continuous Wavelet Transform GUI

In R2011b, you can compute the Fourier transform based continuous wavelet transform (CWT) and inverse CWT using the Wavelet Toolbox graphical user interface wavemenu. To access these graphical tools, enter wavemenu at the command line, and select **Continuous Wavelet 1-D (using FFT)**.

Inverse Continuous Wavelet Transform Using Linear Scales

In R2011b, you can compute the inverse continuous wavelet transform (CWT) for a wider class of analyzing wavelets using icwtlin. icwtlin returns the inverse for CWT coefficients obtained at linearly spaced scales. icwtlin supports the output of cwtft and the output of cwt for a select number of wavelets. See icwtlin for detailed information.

MATLAB Code Generation Support for Denoising and Compression GUIs

In R2011b, you can generate MATLAB code for 1-D and 2-D discrete wavelet transforms (DWT), stationary wavelet transforms (SWT), and wavelet packet transforms. You can denoise or compress a signal or image in the GUI and export the MATLAB code to implement that operation at the command line. This approach allows you to set denoising thresholds or compression ratios aided by visualization tools and save the commands to reproduce those operations at the command line. See Generating MATLAB Code from Wavelet Toolbox GUI for examples.

Signal Reconstruction from Continuous Wavelet Transform Coefficients Demo

R2011b includes a new demo illustrating signal reconstruction using the continuous wavelet transform (CWT). The demo emphasizes the use of the CWT to analyze a signal and reconstruct a time- and scale-based approximation with select coefficients using the inverse CWT. See Signal Reconstruction from Continuous Wavelet Transform Coefficients for details.

Changes in Fourier Transform Based Continuous Wavelet Transform Defaults for Derivative of Gaussian (DOG) and Paul Wavelets

In R2011b, the default values for the smallest scale, scale increment, and number of scales have changed in cwtft for the derivative of Gaussian (DOG) and Paul wavelets.

The change in the defaults also affects the Mexican hat wavelet, which is a special case of the DOG wavelet. In R2011b, the default value of the smallest scale for the Paul and DOG wavelets is `2*dt`, where `dt` is the sampling period. The default scale increment, `ds`, is 0.4875. The default number of scales is `fix(log2(length(sig))/ds)+1` for the Paul wavelet and `max([fix(log2(length(sig))/ds),1])` for the DOG wavelets, where `sig` is the input signal.

Compatibility Considerations

cwtft was introduced in R2011a. In that release, the default smallest scales for the DOG and Paul wavelets are `dt/8` and `dt` respectively, where `dt` is the sampling interval. The default scale increment is 0.5. The default number of scales is `fix(1.5*log2(length(sig))/ds)+1` for the Paul wavelet. For DOG wavelets, the default number of scales is `fix(1.25*log2(length(sig))/ds)+1`, where `sig` is the input signal. You can obtain results in R2011b using `cwtft` with the DOG and Paul wavelets identical to results in R2011a with the default values. To do so, specify the smallest scale, scale increment, and number of scales in a structure or cell array. See cwtft for details.

R2011a

Version: 4.7

New Features

Bug Fixes

Inverse Continuous Wavelet Transform

In R2011a, you can compute the inverse continuous wavelet transform (CWT) using an FFT-based algorithm. The inverse CWT allows you to synthesize approximations to your 1D signal based on selected scales. The inverse CWT is only supported for coefficients obtained using the FFT-based CWT. See icwtft and cwtft for details.

FFT-based Continuous Wavelet Transform

In R2011a, you can compute the continuous wavelet transform (CWT) using an FFT-based algorithm with cwtft. The CWT computed using an FFT algorithm supports the computation of the inverse CWT. See cwtft and icwtft for details. Only select wavelets are valid for use with cwtft. See cwtftinfo for a list of supported wavelets.

Pattern-adapted Wavelets for Signal Detection Demo

In R2011a there is a new demo using pattern adapted wavelets for signal detection. You can view this demo here Pattern adapted wavelets for signal detection. The Wavelet Toolbox software enables you to design admissible wavelets based on the pattern you wish to detect. Designing a valid wavelet based on your desired pattern allows you to exploit the optimality of matched filtering in the framework of the CWT. The demo illustrates this process on simulated data and human EEG recordings.

R2010b

Version: 4.6

New Features

Bug Fixes

Cone of Influence for Continuous Wavelet Transform

In R2010b, you can compute the cone of influence (COI) for the continuous wavelet transform (CWT) of a signal. At each scale, the COI determines the set of CWT coefficients influenced by the value of the signal at a specified position. The COI provides an important visual aid in interpreting the CWT. By overlaying the cone of influence on the CWT image, you can determine which CWT coefficients each value of the signal affects at every scale. See conofinf for details.

Wavelet Cross Spectrum and Coherence

In R2010b, you can estimate the wavelet cross spectrum and wavelet coherence of two time series. The wavelet cross spectrum and coherence provide wavelet-based alternatives for the Fourier-based cross spectrum and coherence. These wavelet estimators are suitable for nonstationary signals. Using a complex-valued analyzing wavelet, you can also examine intervals in the time-scale plane where the two time series exhibit common phase behavior. See wcoher and the new demo Wavelet Coherence for details.

Wavelet Packet Spectrum

In R2010b, you can compute the wavelet packet spectrum with wpspectrum. The wavelet packet spectrum provides a time-frequency analysis of a time series. The wavelet packet spectrum is useful as wavelet-based counterpart of the short-time Fourier transform.

Natural and Frequency Ordering of Wavelet Packet Terminal Nodes

In R2010b, you can order the wavelet packet transform terminal nodes by natural (Payley) or frequency (sequency) order. See otnodes for details.

Image and Signal Approximation Quality Metrics

In R2010b, you can measure the quality of your signal or image approximation using a number of widely-used quality metrics. These metrics include: the peak signal-to-noise ratio (PSNR), the mean square error (MSE), the maximum absolute error, and the energy ratio of the approximation to the original. See measerr for details.

R2010a

Version: 4.5

New Features

Bug Fixes

3-D Discrete Wavelet Transform

This release adds new functions and a GUI to support the 3-D discrete wavelet transform. This new functionality lets you decompose, analyze, and display a 3-D object using a different wavelet for each dimension. The new functions are: dwt3, idwt3, wavedec3, and waverec3. A demo (`wavelet3ddemo`) is also included.

Nondecimated Wavelet Transform

New nondecimated wavelet transform functions support signals of arbitrary size and different extension modes. Previous functionality had two limitations: signal length had to equal a power of 2 and the only allowable extension mode was periodized. The new functions are: ndwt, indwt, ndwt2, and indwt2. A demo (`ndwtdemo`) is also included.

New Denoising Function

The new cmddenoise function uses interval-dependent denoising to compute the denoised signal and coefficients. This allows you to apply different denoising thresholds to different portions of the signal, which is typically nonuniform. You can also export thresholds from the GUI and use them in the `cmddenoise` function. The toolbox includes a denoising demo (`cmddenoise`).

R2009b

Version: 4.4.1

Bug Fixes

R2009a

Version: 4.4

New Features

Bug Fixes

New Demos

The toolbox now includes three new demos:

Adding a New Wavelet

Wavelet Interval-dependant Denoising

Wavelet Scalograms

R2008b

Version: 4.3

New Features

Bug Fixes

True Image Compression Support

The new `wcompress` functions lets you compress 2D image data. You can also interactively compress images using the new Two-Dimensional Images Compression GUI.

New Demo

The toolbox now includes a new codepad demo on image compression.

R2008a

Version: 4.2

New Features

Bug Fixes

True Color Images Support

The toolbox can now process true color images. All major toolbox GUIs and all of the 2D-oriented command line functions have been also updated and support true color images.

New Extension Modes for Continuous Wavelets

The new cwtext function lets you calculate 1D continuious wavelet parameters using extension parameters.

New Norms Calculation

The Multisignal 1D GUI and other related GUIs now include 1-norm, 2-norm, and inf-norm calculations.

Wavelet Families Display

A new function, waveletfamilies, displays all the available wavelet families and their properties.

Single Data Type Support

The swt2 and iswt2 functions now support single data types.

New Demos

The toolbox now includes the following new codepad demos:

- Multiscale Principal Component Analysis
- Multivariate Denoising

R2007b

Version: 4.1

New Features

Bug Fixes

Importing and Exporting between GUIs and Workspace

You can now import data from the workspace to all toolbox GUIs and export data from all toolbox GUIs to the workspace. Use **Import from Workspace** and **Export to Workspace**, respectively, on the GUI's **File** menu.

Scalograms for Continuous Wavelet Transforms

The ability to compute scalograms of the wavelet coefficients in continuous wavelet analysis has been added as an option to the cwt function. You can also pass the structure produced by cwt directly to the new wscalogram function. Scalograms show the percentage of energy in each wavelet coefficient.

Constructing Clusters from Hierarchical Cluster Trees

You can now construct clusters from hierarchical cluster trees in multisignal analysis using the new mdwtcluster function.

R2007a

Version: 4.0

New Features

Bug Fixes

1D Multisignal Analysis, Compression, and Denoising Added

The following command-line functions for 1D multisignal analysis, compression, and denoising have been added to the toolbox:

mswcmp	Multisignal 1D compression using wavelets.
mswcmpscr	Multisignal 1D wavelet compression scores.
mswcmptp	Multisignal 1D compression thresholds and perf.
mswden	Multisignal 1D denoising using wavelets.
mswthresh	Performs Multisignal 1D thresholding.

1D Multisignal Wavelet and Clustering Added

The following command-line functions for 1D multisignal wavelets and clustering have been added to the toolbox:

chgwdeccfs	Change Multisignal 1D decomposition coeffs
mdwtdec	Multisignal 1D wavelet decomposition
mdwtrec	Multisignal 1D wavelet reconstruction.
wdecenergy	Multisignal 1D decomposition energy repartition

Note Clustering analyses require that Statistics Toolbox™ is installed.

Wavelet 1D Multisignal Analysis GUI Added

A graphical user interface for 1-D multisignal analysis has been added. To start this GUI, select **Multisignal Analysis 1D** from the `wavemenu` dialog.

R2006b

Version: 3.1

New Features

Bug Fixes

Multivariate De-noising Added

A new command-line function (wmulden) and a new GUI (**Multivariate Denoising** from the wavemenu initial window) for de-noising a matrix of signals have been added. Both the function and GUI take into account the signals themselves and the correlations between the signals. A two-step process is used. First, a change of basis is performed to deal with noise spatial correlation de-noising in the new basis. Then, a principal component analysis is performed to take advantage of the deterministic relationships between the signals, leading to an additional de-noising effect.

Multiscale Principal Component Analysis Added

A new command-line function (wmspca) and a new GUI (**Multiscale Princ. Comp. Analysis** from the wavemenu initial window) for simplifying a matrix of signals have been added. Both the function and GUI take into account the signals themselves and the correlations between the signals. The multiscale principal component analysis mixes wavelet decompositions and principal component analysis.

New Demos

The following new demos are added:

- Continuous and Discrete Wavelet Analysis
- Detecting Discontinuities and Breakdown Points
- De-Noising Signals and Images
- Data Compression using 2D Wavelet Analysis
- Image Fusion
- Detecting Self-Similarity
- Wavelet Packets: Decomposing the Details

R2006a

Version: 3.0.4

No New Features or Changes

R14SP3

Version: 3.0.3

No New Features or Changes

R14SP2

Version: 3.0.2

No New Features or Changes