Fuzzy Logic Toolbox™ Release Notes

MATLAB®

MathWorks®

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

Fuzzy Logic Toolbox ™ Release Notes

Trademarks

Patents

Contents

R2018a

Fuzzy Logic Controller Block: Troubleshoot FIS evaluation using new diagnostic messages 1-2

New evalfisOptions Option Set: Specify options for evaluating fuzzy systems 1-2

evalfis Function: New diagnostic message behavior 1-3

evalfis Function: Intermediate fuzzy inference outputs for Sugeno systems analogous to outputs for Mamdani systems 1-3

Functionality being removed or changed 1-4

R2017b

Code Generation Improvements: Generate code for single and fixed-point data types, and custom membership and inference functions 2-2

PLC Deployment: Generate IEC 61131-3 Structured Text from fuzzy logic controllers 2-2

Fuzzy Logic Controller Block Improvements: Configure additional block parameters, and access intermediate fuzzy inference results . . . . 2-2 evalfis Command: Evaluate FIS output variable ranges over a smaller number of sample points 2-3

R2017a

Unified genfis Command: Generate fuzzy inference system structures using a single command 3-2

anfisOptions Command: Specify options for training adaptive neuro-fuzzy inference systems 3-4

gensurfOptions Command: Specify options for generating fuzzy inference system output surfaces 3-5

newfis Command: Specify options using Name,Value pairs . . . 3-6

parsrule Command: Specify options using Name,Value pairs 3-6

showrule Command: Specify options using Name,Value pairs 3-7

subclust Command: Specify options using Name,Value pairs 3-8

Obtain fuzzy inference system properties using improved getfis command 3-8

Functionality being removed or changed 3-9

R2016b

Standalone Applications for ANFIS Training: Deploy neuro-adaptive fuzzy inference code using MATLAB Compiler . . . 4-2

R2016a

Bug Fixes

R2015b

Bug Fixes

R2015a

Bug Fixes

R2014b

Commands to open Fuzzy Logic Designer and Neuro-Fuzzy Designer renamed 8-2

R2014a

Example that shows how to use a fuzzy inference system to detect edges in an image 9-2

R2013b

Bug Fixes

R2013a

No New Features or Changes

R2012b

No New Features or Changes

R2012a

No New Features or Changes

R2011b

No New Features or Changes

R2011a

No New Features or Changes

R2010b

No New Features or Changes

R2010a

No New Features or Changes

R2009b

No New Features or Changes

R2009a

No New Features or Changes

R2008b

No New Features or Changes

R2008a

No New Features or Changes

R2007b

New Demo 22-2

R2007a

No New Features or Changes

R2006b

No New Features or Changes

R2006a

No New Features or Changes

R14SP3

No New Features or Changes

R14SP2

No New Features or Changes

1

R2018a

Version: 2.3.1

New Features

Bug Fixes

Compatibility Considerations

Fuzzy Logic Controller Block: Troubleshoot FIS evaluation using new diagnostic messages

The Fuzzy Logic Controller block now reports potential problems that can produce unexpected output values during a simulation. The block generates diagnostic messages for the following conditions:

- Input values outside of their specified variable ranges
- No rules fired for a given output at the current input values
- Empty output fuzzy sets

You can specify whether these diagnostic messages are reported as warnings, reported as errors, or ignored.

For more information, see Fuzzy Logic Controller.

New evalfisOptions Option Set: Specify options for evaluating fuzzy systems

You can now specify options for evaluating fuzzy systems at the command line using the new `evalfisOptions` option set. Using this option set, you can specify:

- The number of sample points to use when evaluating output fuzzy sets.
- Whether diagnostic messages for potential problems are reported as warnings, reported as errors, or ignored.

For more information, see `evalfisOptions` and `evalfis`.

Compatibility Considerations

Previously, when evaluating a FIS using `evalfis`, you specified the number of sample points in output fuzzy sets using the optional input argument `numPts`.

```
output = evalfis(fis,input,numPts);
```

Starting in R2018a, modify your code to use an `evalfisOptions` option set.

```
opt = evalfisOptions('NumSamplePoints',numPts);
output = evalfis(fis,input,opt);
```

evalfis Function: New diagnostic message behavior

You can now control how the `evalfis` function reports potential problems that can produce unexpected output values during a simulation. The function generates diagnostic messages for the following conditions:

- Input values outside of their specified variable ranges
- No rules fired for a given output at the current input values
- Empty output fuzzy sets

You can specify whether these diagnostic messages are reported as warnings, reported as errors, or ignored. To do so, specify the corresponding options in an `evalfisOptions` option set.

For more information, see `evalfisOptions` and `evalfis`.

Compatibility Considerations

Previously, the `evalfis` function had the following behaviors for diagnostic conditions.

Diagnostic Condition	Previous Behavior
Input values outside of their specified variable ranges	MATLAB® warning
No rules fired for a given output at the current input values	MATLAB Command Window message
Empty output fuzzy sets	MATLAB Command Window message

Starting in R2018a, these diagnostic conditions are reported as MATLAB warnings by default. You can change this behavior by specifying the corresponding options in an `evalfisOptions` option set.

evalfis Function: Intermediate fuzzy inference outputs for Sugeno systems analogous to outputs for Mamdani systems

When evaluating a Sugeno system using the following syntax, the intermediate fuzzy inference results are now analogous to the intermediate results for Mamdani systems.

```
[output,fuzzifiedInputs,ruleOutputs,aggregatedOutput] = evalfis(input,fis);
```

For a Sugeno system:

- `ruleOutputs` now returns an array that contains the scalar output value for each rule; that is, the product of the rule firing strength and the rule output level.
- `aggregatedOutput` now returns the sum of all the rule output values for each output variable.

For more information, see `evalfis`.

Compatibility Considerations

Previously, for a Sugeno fuzzy system:

- `ruleOutputs` returned an array that contained the output level for each rule.
- `aggregatedOutput` returned an array that contained the firing strength for each rule.

Starting in R2018a, if your code returns intermediate fuzzy inference results when evaluating a Sugeno system using `evalfis`, modify your code to use the new `ruleOutputs` and `aggregatedOutput` results.

Functionality being removed or changed

Functionality	**Result**	**Use This Instead**	**Compatibility Considerations**

`output = evalfis(fis,input,numPts);`	Still works	`opt = evalfisOptions('NumSamplePoints',numPts);`
		`output = evalfis(fis,input,opt);`
			To specify the number of sample points for output fuzzy sets, use an `evalfisOptions` option set. For more information, see “New evalfisOptions Option Set: Specify options for evaluating fuzzy systems” on page 1-2.

Functionality	Result	Use This Instead	Compatibility Considerations
Diagnostic messages when evaluating fuzzy systems using `evalfis`	Still works	Not applicable	By default, diagnostic messages are now reported as warnings. To change this behavior, specify the corresponding options in an `evalfisOptions` option set. For more information see, "evalfis Function: New diagnostic message behavior" on page 1-3.
`[output,fuzzifiedInputs,ruleOutputs,aggregatedOutput] = evalfis(input,fis)` when evaluating Sugeno systems	Still works	Not applicable	The behaviors of the `ruleOutputs` and `aggregatedOutput` argument of `evalfis` have changed for evaluating Sugeno systems. For more information, see "evalfis Function: Intermediate fuzzy inference outputs for Sugeno systems analogous to outputs for Mamdani systems" on page 1-3.

2

R2017b

Version: 2.3

New Features

Bug Fixes

Code Generation Improvements: Generate code for single and fixed-point data types, and custom membership and inference functions

The Fuzzy Logic Controller block now supports code generation for fuzzy systems using:

- Single-precision data.
- Fixed-point data. To generate code for fixed-point data, you need Fixed-Point Designer™ software.
- Custom membership functions and custom inference functions. For more information on specifying custom functions for a fuzzy system, see Build Fuzzy Systems Using Custom Functions.

You can generate code using either Simulink® Coder™ or Simulink PLC Coder™ software.

PLC Deployment: Generate IEC 61131-3 Structured Text from fuzzy logic controllers

The Fuzzy Logic Controller block now supports generation of IEC 61131-3 Structured Text for PLC deployment using Simulink PLC Coder software.

Fuzzy Logic Controller Block Improvements: Configure additional block parameters, and access intermediate fuzzy inference results

For the Fuzzy Logic Controller block, you can now:

- Use a double-precision, single-precision, or fixed-point data type.
- Specify the number of sample points for evaluating the output range of a Mamdani system.
- Access intermediate fuzzy inference results using new optional output ports.

For more information, see Fuzzy Logic Controller.

evalfis Command: Evaluate FIS output variable ranges over a smaller number of sample points

You can now specify the number of sample points for evaluating the output range of a Mamdani fuzzy inference system at the command line as any value greater than `1`. Previously, the minimum value was `101`.

This change applies to the:

- `numPts` input argument of `evalfis`.
- `NumSamplePoints` property of `gensurfOptions`.

3

R2017a

Version: 2.2.25

New Features

Bug Fixes

Compatibility Considerations

Unified genfis Command: Generate fuzzy inference system structures using a single command

The commands for generating the structure of a fuzzy inference system have been unified into a single `genfis` command, which you configure using a new `genfisOptions` option set.

Starting in R2017a, to generate a FIS structure, first create a default `genfisOptions` option set, specifying one of the following structure generation algorithms:

- Grid partitioning

  ```
  opt = genfisOptions('GridPartition');
  ```
- Subtractive clustering

  ```
  opt = genfisOptions('SubtractiveClustering');
  ```
- Fuzzy c-means clustering

  ```
  opt = genfisOptions('FCMClustering');
  ```

You can then modify the options using dot notation. Any options you do not modify remain at their default values.

Compatibility Considerations

Previously, to generate FIS structures, you used the `genfis1`, `genfis2`, or `genfis3` commands with optional input arguments.

These commands may be removed in a future release and, starting in R2017a, using these commands generates a warning. If your code uses `genfis1`, `genfis2`, or `genfis3`, modify the code to use the `genfis` command, specifying options using a `genfisOptions` option set.

Algorithm	Old Syntax	New Syntax

Grid partitioning	`fis = genfis1(data,numMFs,inmftype,outmftype);`	`opt = genfisOptions('GridPartition');`
		`opt.NumMembershipFunctions = numMFs;`
		`opt.InputMembershipFunctionType = inmftype;`
		`opt.OutputMembershipFunctionType = outmftype`
		`inputData = data(:,end-1);`
		`outputData = data(:,end);`
		`fis = genfis(inputData,outputData,opt);`

Subtractive clustering	`fis = genfis2(inputData,outputData,radii,xBounds,options,userCenters);`	`opt = genfisOptions('SubtractiveClustering'`
		`opt.ClusterInfluenceRange = radii;`
		`opt.DataScale = xBounds;`
		`opt.SquashFactor = options(1);`
		`opt.AcceptRatio = options(2);`
		`opt.RejectRatio = options(3);`
		`opt.Verbose = options(4);`
		`opt.CustomClusterCenters = userCenters;`
		`fis = genfis(inputData,outputData,opt);`

FCM clustering	`fis = genfis3(inputData,outputData,type,cluster_n,fcmoptions);`	`opt = genfisOptions('FCMClustering');`
		`opt.FISType = type;`
		`opt.NumClusters = cluster_n;`
		`opt.Exponent = fcmoptions(1);`
		`opt.MaxNumIteration = fcmoptions(2);`
		`opt.MinImprovement = fcmoptions(3);`
		`opt.Verbose = fcmoptions(4);`
		`fis = genfis(inputData,outputData,opt);`

The syntaxes in this table assume that you are specifying all the options for each algorithm. Since the initial `genfisOptions` option set contains default algorithm options, you have to specify only nondefault options. For example, create an FIS using FCM clustering with three clusters, leaving all other options at their default values.

```
opt = genfisOptions('FCMClustering');
opt.NumClusters = 3;
fis = genfis(Xin,Xout,opt);
```

anfisOptions Command: Specify options for training adaptive neuro-fuzzy inference systems

To specify options for training adaptive neuro-fuzzy inference systems using the `anfis` command, you now create an `anfisOptions` option set. You can then modify the options using dot notation. Any options you do not modify remain at their default values.

Compatibility Considerations

Previously, to train an adaptive neuro-fuzzy inference system using `anfis`, you specified the training options using optional input arguments.

```
fis = anfis(trnData,initFIS,trnOpt,dispOpt,chkData,optMethod);
```

Starting in R2017a, if your code uses `anfis`, modify the code to use an `anfisOptions` option set.

```
opt = anfisOptions;
opt.InitialFIS = 3;
fis = anfis(trnData,opt);
```

The following table shows the mapping of the old `anfis` input arguments to the new `anfisOptions` option set.

Old `anfis` Input Argument	New `anfisOptions` Option
`initFIS`	`InitialFIS`
`trnOpt(1)`	`EpochNumber`
`trnOpt(2)`	`ErrorGoal`
`trnOpt(3)`	`InitialStepSize`
`trnOpt(4)`	`StepSizeDecreaseRate`
`trnOpt(5)`	`StepSizeIncreaseRate`
`dispOpt(1)`	`DisplayANFISInformation`
`dispOpt(2)`	`DisplayErrorValues`
`dispOpt(3)`	`DisplayStepSize`
`dispOpt(4)`	`DisplayFinalResults`
`chkData`	`ValidationData`

Old anfis Input Argument	New anfisOptions Option
optMethod	OptimizationMethod

gensurfOptions Command: Specify options for generating fuzzy inference system output surfaces

To specify options for generating fuzzy inference system output surfaces using the `gensurf` command, you now create a `gensurfOptions` option set. You can then modify the options using dot notation. Any options you do not modify remain at their default values.

Compatibility Considerations

Previously, to generate an output surface for a fuzzy inference system using `gensurf`, you specified the generation options using optional input arguments.

```
gensurf(fis,inputs,output,grids,refInput,points);
```

Starting in R2017a, if your code uses `gensurf`, modify the code to use a `gensurfOptions` option set.

```
opt = gensurfOptions;
opt.InputIndex = [1 3];
fis = gensurf(fis,opt);
```

The following table shows the mapping of the old `gensurf` input arguments to the new `gensurfOptions` option set.

Old gensurf Input Argument	New gensurfOptions Option
inputs	InputIndex
output	OutputIndex
grids	NumGridPoints
refinput	ReferenceInputs
points	NumSamplePoints

newfis Command: Specify options using Name,Value pairs

To specify options for creating new fuzzy inference systems using the `newfis` command, you now use `Name,Value` pair arguments. Any `Name,Value` pair arguments that you do not specify remain at their default values.

Compatibility Considerations

Previously, you specified options for the `newfis` command using optional input arguments.

```
fis = newfis('My FIS',fisType,andMethod,orMethod,impMethod,aggMethod,defuzzMethod);
```

Starting in R2017a, if your code uses `newfis`, modify the code to use one or more `Name,Value` pair arguments. For example, create a Mamdani FIS with default options.

```
fis = newfis('My FIS','FISType','mamdani');
```

The following table shows the mapping of the old input arguments to the new `Name,Value` pair arguments.

Old newfis Input Argument	New Name,Value Argument
`fisType`	`'FISType'`
`andMethod`	`'AndMethod'`
`orMethod`	`'OrMethod'`
`impMethod`	`'ImplicationMethod'`
`aggMethod`	`'AggregationMethod'`
`defuzzMethod`	`'DefuzzificationMethod'`

parsrule Command: Specify options using Name,Value pairs

To specify options for creating new fuzzy inference systems using the `parsrule` command, you now use `Name,Value` pair arguments. Any `Name,Value` pair arguments that you do not specify remain at their default values.

Compatibility Considerations

Previously, you specified options for the parsrule command using optional input arguments ruleFormat and lang.

```
outFIS = parsrule(inFIS,ruleList,ruleFormat,lang);
```

Starting in R2017a, if your code uses newfis, modify the code to use one or more Name,Value pair arguments. For example, add a list of rules in 'symbolic' format.

```
fis = parsrule(inFIS,ruleList,'Format','symbolic');
```

The following table shows the mapping of the old input arguments to the new Name,Value pair arguments.

Old parsrule Input Argument	New Name,Value Argument
ruleFormat	'Format'
lang	'Language'

showrule Command: Specify options using Name,Value pairs

To specify options for viewing the rules of a fuzzy inference system using the showrule command, you now use Name,Value pair arguments. Any Name,Value pair arguments that you do not specify remain at their default values.

Compatibility Considerations

Previously, you specified options for the showrule command using optional input arguments indexList, format, and lang.

```
showrule(fis,indexList,format,lang);
```

Starting in R2017a, if your code uses newfis, modify the code to use one or more Name,Value pair arguments. For example, view the first fuzzy rule in fis.

```
showrule(fis,'RuleIndex',1);
```

The following table shows the mapping of the old input arguments to the new Name,Value pair arguments.

Old showrule Input Argument	New Name,Value Argument
indexList	'RuleIndex'
format	'Format'
lang	'Language'

subclust Command: Specify options using Name,Value pairs

To specify options for subtractive clustering using the subclust command, you now use Name,Value pair arguments. Any Name,Value pair arguments that you do not specify remain at their default values.

Compatibility Considerations

Previously, you specified options for the subclust command using optional input arguments xBounds and options.

```
fisOut = subclust(fisIn,radii,xBounds,options);
```

Starting in R2017a, if your code uses newfis, modify the code to use one or more Name,Value pair arguments. For example, specify clustering options.

```
fisOut = subclust(fisIn,radii,'Options',options);
```

The following table shows the mapping of the old input arguments to the new Name,Value pair arguments.

Old subclust Input Argument	New Name,Value Argument
xBounds	'DataScale'
options	'Options'

Obtain fuzzy inference system properties using improved getfis command

Several getfis syntaxes that previously printed formatted properties to the Command Window and also returned properties now perform a single action.

- getfis(fis) now just prints FIS properties to the Command Window.

- `getfis(fis,vartype,varindex)` now just returns variable properties in a structure.
- `getfis(fis,vartype,varindex,'mf',mfIndex)` now just returns membership function properties in a structure.

Compatibility Considerations

Starting in R2017a, the following `getfis` syntaxes have a new behavior.

Syntax	Previous Behavior	New Behavior
`getfis(fis)`	Print formatted list of FIS properties to Command Window, and return FIS name.	Print formatted list of FIS properties to Command Window.
`getfis(fis,varType,varIndex)`	Print formatted list variable properties to Command Window, and return structure that contains variable properties.	Return structure that contains variable properties.
`getfis(fis,varType,varIndex,'mf',mfIndex)`	Print formatted list membership function properties to Command Window, and return structure that contains membership function properties.	Return structure that contains membership function properties.

Functionality being removed or changed

Functionality	Result	Use This Instead	Compatibility Considerations
`genfis1`, `genfis2`, and `genfis3` commands	Warns	`genfis` command	See "Unified genfis Command: Generate fuzzy inference system structures using a single command" on page 3-2.
Specify optional input arguments for the `anfis` command	Still works	Specify options using `anfisOptions` command.	See "anfisOptions Command: Specify options for training adaptive neuro-fuzzy inference systems" on page 3-4.

Functionality	Resul t	Use This Instead	Compatibility Considerations
Specify optional input arguments for the `gensurf` command	Still works	Specify options using `gensurfOptions` command.	See "gensurfOptions Command: Specify options for generating fuzzy inference system output surfaces" on page 3-5.
Specify optional input arguments for `newfis` command	Still works	Specify options using `Name,Value` pair arguments.	See "newfis Command: Specify options using Name,Value pairs" on page 3-6.
Specify optional input arguments for `parsrule` command	Still works	Specify options using `Name,Value` pair arguments.	See "parsrule Command: Specify options using Name,Value pairs" on page 3-6.
Specify optional input arguments for `showrule` command	Still works	Specify options using `Name,Value` pair arguments.	See "showrule Command: Specify options using Name,Value pairs" on page 3-7.
Specify optional input arguments for `subclust` command	Still works	Specify options using `Name,Value` pair arguments.	See "subclust Command: Specify options using Name,Value pairs" on page 3-8.
`getfis` command syntaxes that both print and return properties	Still works	Syntaxes now either print or return properties, not both.	See "Obtain fuzzy inference system properties using improved getfis command" on page 3-8.

4

R2016b

Version: 2.2.24

New Features

Bug Fixes

Standalone Applications for ANFIS Training: Deploy neuro-adaptive fuzzy inference code using MATLAB Compiler

The `anfis` command now supports application deployment using MATLAB Compiler™. For more information on building and deploying standalone applications from MATLAB programs, see MATLAB Compiler.

R2016a

Version: 2.2.23

Bug Fixes

6

R2015b

Version: 2.2.22

Bug Fixes

R2015a

Version: 2.2.21

Bug Fixes

8

R2014b

Version: 2.2.20

New Features

Bug Fixes

Commands to open Fuzzy Logic Designer and Neuro-Fuzzy Designer renamed

`fuzzy` is renamed to `fuzzyLogicDesigner`. Use this command to open the Fuzzy Logic Designer app.

`anfisedit` is renamed to `neuroFuzzyDesigner`. Use this command to open the Neuro-Fuzzy Designer app.

9

R2014a

Version: 2.2.19

New Features

Bug Fixes

Example that shows how to use a fuzzy inference system to detect edges in an image

The Fuzzy Logic Image Processing example shows how to use a fuzzy inference system to detect edges in an image.

R2013b

Version: 2.2.18

Bug Fixes

11

R2013a

Version: 2.2.17

No New Features or Changes

R2012b

Version: 2.2.16

No New Features or Changes

R2012a

Version: 2.2.15

No New Features or Changes

R2011b

Version: 2.2.14

No New Features or Changes

R2011a

Version: 2.2.13

No New Features or Changes

R2010b

Version: 2.2.12

No New Features or Changes

R2010a

Version: 2.2.11

No New Features or Changes

18

R2009b

Version: 2.2.10

No New Features or Changes

R2009a

Version: 2.2.9

No New Features or Changes

20

R2008b

Version: 2.2.8

No New Features or Changes

21

R2008a

Version: 2.2.7

No New Features or Changes

R2007b

Version: 2.2.6

New Features

New Demo

Fuzzy Logic Toolbox software has a new demo Fuzzy C-Means Clustering for Iris Data, which illustrates the use of Fuzzy C-Means clustering for Iris dataset.

23

R2007a

Version: 2.2.5

No New Features or Changes

24

R2006b

Version: 2.2.4

No New Features or Changes

R2006a

Version: 2.2.3

No New Features or Changes

R14SP3

Version: 2.2.2

No New Features or Changes

R14SP2

Version: 2.2.1

No New Features or Changes