

Spline Toolbox Release Notes

Contents

Summary by Version	4
Version 3.3 (R2006a) Spline Toolbox	7
Version 3.2.2 (R14SP3) Spline Toolbox	11
Version 3.2.1(R14SP1) Spline Toolbox	13
Version 3.2 (R13+) Spline Toolbox	14
Version 3.1.1 (R13) Spline Toolbox	18
Version 3.0 (R12) Spline Toolbox	21
Compatibility Summary for Spline Toolbox	26

Summary by Version

This table provides quick access to what's new in each version. For clarification, see About Release Notes.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V3.3 (R2006a)	Yes Details	Yes Summary	No bug fixes	Printable Release Notes: PDF V3.3 product documentation
V3.2.2 (R14SP3)	Yes Details	Yes Summary	No bug fixes	No
V3.2.1 (R14SP1)	No	No	No bug fixes	No
V3.2 (R13+)	Yes Details	Yes Summary	No bug fixes	No
V3.1.1 (R13)	Yes Details	Yes Summary	Fixed bugs	No
V3.0 (R12)	Yes Details	Yes Summary	List of bug fixes	No

About Release Notes

Use release notes when upgrading to a newer version to learn about new features and changes, and the potential impact on your existing files and practices. Release notes are also beneficial if you use or support multiple versions.

If you are not upgrading from the most recent previous version, review release notes for all interim versions, not just for the version you are installing. For example, when upgrading from V1.0 to V1.2, review the New Features and Changes, Version Compatibility Considerations, and Bug Reports for V1.1 and V1.2.

New Features and Changes

These include

- New functionality
- Changes to existing functionality
- Changes to system requirements (complete system requirements for the current version are at the MathWorks Web site)
- Any version compatibility considerations associated with each new feature or change

Version Compatibility Considerations

When a new feature or change introduces a known incompatibility with the previous version, its description includes a **Compatibility Considerations** subsection that details the impact. For a list of all new features and changes that have compatibility impact, see the Compatibility Summary for Spline Toolbox.

Compatibility issues that become known after the product has been released are added to Bug Reports at the MathWorks Web site. Because bug fixes can sometimes result in incompatibilities, also review fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks Bug Reports is a user-searchable database of known problems, workarounds, and fixes. The MathWorks updates the Bug Reports database as new problems and resolutions become known, so check it as needed for the latest information.

Access Bug Reports at the MathWorks Web site using your MathWorks Account. If you are not logged in to your MathWorks Account when you link to Bug Reports, you are prompted to log in or create an account. You then can view bug fixes and known problems for R14SP2 and more recent releases.

The Bug Reports database was introduced for R14SP2 and does not include information for prior releases. You can access a list of bug fixes made in prior versions via the links in the summary table.

Related Documentation at Web Site

Printable Release Notes (PDF). You can print release notes from the PDF version, located at the MathWorks Web site. The PDF version does not support links to other documents or to the Web site, such as to Bug Reports. Use the browser-based version of release notes for access to all information.

Product Documentation. At the MathWorks Web site, you can access complete product documentation for the current version and some previous versions, as noted in the summary table.

Version 3.3 (R2006a) Spline Toolbox

This table summarizes what's new in Version 3.3 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports at Web site	Printable Release Notes: PDF V3.3 product documentation

New features and changes introduced in this version are

- Controlled Extrapolation Beyond the Basic Interval
- Interpolating Planar Tangent-Continuous Piecewise Biarc Curve
- rsmak Now Provides a Torus As Well As a Longer Arc
- rsmak Input and Output For Getting an Arc Is Changed
- Part 'k...' For a Ppform Disallowed
- Strange Error Message from fnval Eliminated
- Changes to Spline Tool

Controlled Extrapolation Beyond the Basic Interval

When a spline in ppform is evaluated at some site outside its basic interval, the value at that site of the 'nearest' polynomial piece is returned. On the other hand, when a spline in B-form is evaluated outside its basic interval, the value 0 is returned. These are the two extreme ways of extrapolating a spline beyond its basic interval. Assuming the spline to be of order k , we use, in the first case, a polynomial of order k that matches k -fold the spline at the relevant endpoint of its basic interval, and this we call extrapolation of order k . In the second case, we use a polynomial of order 0 (i.e., the zero polynomial) that matches the spline 0-fold (i.e., perhaps not at all) at the relevant endpoint of its basic interval, and this we call extrapolation of order 0.

The new toolbox function, `fnxtr(f, j)`, returns a spline in `ppform` that extrapolates the spline in `f` to order `j`, with 2 the default value for `j` in case `j` is not explicitly given. This new feature is of immediate use for the evaluation of smoothing splines since, by its definition, a smoothing spline of order $2m$ is to be extrapolated beyond its basic interval to order m , not 0 nor $2m$.

A new example, Least-Squares Approximation by “Natural” Cubic Splines in the Spline Toolbox documentation makes essential use of this new feature.

For more information, see the `fnxtr` reference page.

Interpolating Planar Tangent-Continuous Piecewise Biarc Curve

The new toolbox function, `rscvn(p, u)`, provides the planar tangent-continuous piecewise biarc curve that passes, in order, through the given points $p(:, 1)$, $p(:, 2)$, ..., $p(:, n)$ in the plane and, for all j , is, at the point $p(:, j)$, perpendicular to the vector $u(:, j)$. Default values for the vectors $u(:, j)$ are used when the user does not supply them.

`rscvn` makes use of the fact that the Spline Toolbox can handle rational splines, hence can represent exactly curves made up of circular arcs. The particular rational spline used here is piecewise quadratic, with two pieces between neighboring data points, hence the term *biarc*. Because of the geometry of the construction used, users need not provide parameter values associated with the given data points, as they would have to if they wanted to use `csapi` or `spapi` for the construction of a smooth interpolating curve.

An added example, in the Spline Toolbox demo “More Spline Curves”, illustrates what may have been a very early use of a tangent-continuous piecewise circular arc curve.

For more information, see the `rscvn` reference page.

rsmak Now Provides a Torus As Well As a Longer Arc

`rsmak` now provides the `rBform` of a torus for which you can specify the radii of the outer and the inner circle (of which it is, in a way, the tensor product). Precisely,

```
rsmak('torus', radius, ratio)
```


returns a description of the torus centered at the origin and with the z-axis its axis whose major circle has the specified radius, and whose minor circle has the radius `radius*ratio`.

Also,

```
rsmak('arc',radius,center,[alpha,beta])
```

returns the circular arc, of the given radius and the given center, that starts at angle `alpha` and ends at angle `beta`. If the difference between `alpha` and `beta` is (in absolute value) greater than 2π , the arc includes a full circle.

rsmak Input and Output For Getting an Arc Is Changed

Getting an arc of specified extent used to require the following command:

```
rsmak('arc',radius,center,alpha,beta)
```

in which the beginning angle, `alpha`, and end angle, `beta`, for the arc are specified by two separate input arguments. These two arguments are now combined into one 2-vector argument, as follows:

```
rsmak('arc',radius,center,[alpha,beta])
```

Further, the arc now produced starts at angle `alpha` and ends at angle `beta` and may well cover an entire circle and even more than one. Formerly, the arc produced would never exceed a half circle.

Compatibility Consideration

You are encouraged to change to the new `rsmak` syntax. The earlier syntax, which required separate arguments for `alpha` and `beta`, continues to be valid but may be removed in a future release.

Part 'k...' For a Ppform Disallowed

The request `fnbrk(f,part)`, for the part `'k...'` of a `ppform` now produces an error message.

Compatibility Consideration

The request `fnbrk(f,part)`, with `part` a string starting with `'k'`, used to return the order of the spline in `f` in case the spline is in `ppform`, while it

returns the knot sequence of that spline in case the spline is in B-form. As this can lead to confusion when a user forgets whether the spline of interest is in ppform or B-form, and the former response was undocumented anyway, asking for the part 'k...' of a ppform now produces an error message.

Strange Error Message from fnval Eliminated

Specifying the evaluation sites for `fnval` as a column cell array does not automatically result in an error message.

Compatibility Consideration

When the evaluation sites for `fnval` are specified by a column cell array rather than the required row cell array, a mysterious error message would result. In this situation, `fnval` now tries to make sense of the input by changing the column cell array to a row cell array and give an error message only if the length of the resulting row cell array is incompatible with the number of variables for the function being evaluated.

Changes to Spline Tool

- Proposed names under which to save the data are now serialized.
- The proposed increments/decrements in a knot's location to be used now change suitably when the number of knots changes.
- The legend in the main graph stays put when an approximation is renamed.
- A point highlighted in the main graph now always shows in the visible part of the data point list.

Version 3.2.2 (R14SP3) Spline Toolbox

This table summarizes what's new in Version 3.2.2 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details listed under Compatibility Considerations , below. See also Summary	Bug Reports at Web site	No

New features and changes introduced in this version are

- rsmak Now Provides an Arc
- Changes to Spline Tool
- Compatibility Considerations

rsmak Now Provides an Arc

rsmak now provides the rBform of an arc for which you can specify the center, radius, and extent.

rsmak(object, varargin) returns the specific geometric shape specified by the string object. For example,

```
rsmak('arc', radius, center, alpha, beta)
```

provides a quadratic rational that describes the arc of the given radius (default 1) and center (default (0,0)) of total arc length $\leq \text{radius} \cdot \pi$ that covers the angle interval between the given alpha (default 0) and beta (default pi).

Changes to Spline Tool

- The last of the examples offered during startup now starts, appropriately, with spline interpolation rather than cubic spline interpolation.

- Deleting the Spline Tool window after a spline approximation has been generated now brings up a request for confirmation.
- During longer calculations, the cursor now becomes the hour glass symbol to inform the user that the Spline Tool is working.

Compatibility Considerations

B-form Evaluation Outside the Basic Interval Returns 0

Evaluation of a multivariate spline in B-form at scattered sites now conforms to the definition that such a spline is zero outside its basic interval. `fnval` now returns the value 0 at any site outside the basic interval of a spline in B-form.

Although this has been the documented behavior, it was not previously enforced.

spaps Handles Vector-Valued Data Correctly in the Case of Zero Tolerance.

For the smoothing spline function, `spaps`, if you provide vector-valued data and specify a tolerance of 0 as a way of interpolating the data, `spaps` now handles the data correctly. Prior to this change, `spaps` produced an error message.

fncmb Returns Error if Only One Coefficient Array Is a Scalar

`fncmb` now performs stricter input checking to enforce that the two splines `f` and `g` in the command `fncmb(f,g)` have the same order and the same knot or break sequences. Previously, if one but not the other of the coefficient arrays of the splines `f` or `g` was a scalar, `fncmb` may have produced incorrect results or a confusing error message.

Version 3.2.1(R14SP1) Spline Toolbox

This table summarizes what's new in Version 3.2.1 (R14SP1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
No	No	No bug fixes	No

This version includes only minor bug fixes.

Version 3.2 (R13+) Spline Toolbox

This table summarizes what's new in Version 3.2 (R13+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details listed under Compatibility Considerations , below. See also Summary	No bug fixes	No

New features and changes introduced in this version are

- ND-Valued Data and Spline Functions
- Repeated Sites
- Updated Demos
- Toolbox Function Summary
- Compatibility Considerations

ND-Valued Data and Spline Functions

All *ap* toolbox functions now accept matrix- and ND-valued data. These functions are: csape, csapi, csaps, spap2, spapi, spaps, tpaps. All toolbox fn* functions can now handle the resulting matrix- or ND-valued spline functions.

Repeated Sites

All *ap* toolbox functions now accept data with repeated sites, hence so does the GUI `splinetool`. Specifically,

<code>csape</code> , <code>csapi</code>	Replace data points having the same site by their average.
<code>csaps</code> , <code>spaps</code> , <code>spap2</code> , <code>tpaps</code>	Replace data points with the same site by their weighted average, with its weight the average of the corresponding weights. These functions are concerned with smoothing or least-squares approximation.
<code>spaps</code>	Adjusts the given tolerance in order to make up for the resulting change in the value of the error measure.
<code>spapi</code>	Continues to interpret data with repeated sites as a request for osculatory interpolation

Updated Demos

Several demos have been enlarged to include additional illustrations and examples. You can now access the Spline Toolbox demos from the **Demos** tab in the Help browser, or you can run them by typing `playshow demoname` at the command line.

Toolbox Function Summary

Version 3.2 of the Spline Toolbox provides the following:

- “New Functions” on page -15
- “Functions with New or Changed Capabilities” on page -16

New Functions

Function	Purpose
<code>fnchg</code>	A new toolbox function <code>fnchg(fn, part, value)</code> enables you to change the target dimension or the basic interval of the form in <code>fn</code> .

Functions with New or Changed Capabilities

Function	Purpose
csape	<p>Input of specific end conditions for the univariate case is now the same as for the multivariate case. In both cases, you now supply specific values for endpoint conditions as additional data values. See the reference page for details.</p> <p>In earlier versions, you used a fourth argument, <code>valconds</code>, to supply specific values for endpoint conditions in the univariate case. This argument remains valid for backward compatibility, but it may be removed in a future release, and you are encouraged to discontinue its use.</p>
fnbrk	<p>An expanded syntax, <code>[out1, ..., outn] = fnbrk(f, part1, ..., partm)</code>, returns the part(s) of the form in <code>f</code> specified by <code>part1, ..., partn</code>, where $n \leq m$. These are the parts used when the form was put together, in <code>spmak</code>, <code>ppmak</code>, <code>rpmak</code>, <code>rsmak</code>, or <code>stmak</code>, but also other parts derived from these.</p>
fnval	<p>You can now use a multidimensional array to specify the sites at which <code>fnval</code> is to evaluate a spline function.</p>
spap2, spapi	<p>Now require that, in case of data <code>x,y</code> with <code>y</code> an array, the data value corresponding to <code>x(j)</code> is <code>y(:, j)</code>, not <code>y(j, :)</code>. See “<code>spap2</code>, <code>spapi</code> Input Requirements Updated” on page -17 for more information.</p>
splinetool	<p>From the new Tools menu, you can now toggle the grid and the legend. From the Help menu, you can also look up short descriptions of technical terms used.</p>

Compatibility Considerations

NaNs and Infs

Prior to Version 3.2, NaNs or Infs in the given data would propagate in the normal way through the calculations, leading to NaNs and/or Infs in the numbers that make up the output.

Now, any data point containing NaN or Inf is ignored, but a warning to that effect is printed in the command window. This affects all **ap** commands, i.e., *csape*, *csapi*, *csaps*, *spap2*, *spapi*, *spaps*, and *tpaps*.

csape Input Simplified

In the multivariate case, you supply specific values for endpoint conditions as additional data values. Starting with Version 3.2, you should use the same scheme in the univariate case.

In earlier versions, you used a fourth argument, *valconds*, to supply specific values for endpoint conditions in the univariate case. This argument remains valid for backward compatibility, but it may be removed in a future release, and you are encouraged to discontinue its use.

spap2, spapi Input Requirements Updated

For the sake of uniformity with other **ap** commands, *spap2(knots, order, x, y)* and *spapi(knots, x, y)* now require that, when *y* is an array, *y* must have as many columns as there are data sites in the vector *x*. That is, the data value corresponding to *x(j)* is *y(:, j)*, not *y(j, :)*. This change originally appeared in Version 3.2 Release 13+.

Prior to Version 3.2, if *x* was a column vector, and *y* was an array, then *spap2* and *spapi* would take *y(j, :)*, rather than *y(:, j)*, to be the data value at *x(j)*.

Version 3.1.1 (R13) Spline Toolbox

This table summarizes what's new in Version 3.1.1 (R13):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details listed under Compatibility Consideration , below. See also Summary	Fixed bugs	No

New features and changes introduced in this version are

- Smoothing Splines for Scattered Bivariate Data
- splinetool Command History M-File
- Spline-Specific Zero Finder
- Spline-Specific Minimum Finder
- Function Summary
- Compatibility Consideration

Smoothing Splines for Scattered Bivariate Data

The new command `tpaps` enables you to create thin-plate spline approximations f that satisfy, approximately or exactly, the equation $z = f(x, y)$ for given data values z at given scattered data sites (x, y) in the plane. The associated collocation matrix is provided by `stcol`. You can also create the first-order derivatives of a thin-plate spline.

The spline created by `tpaps` is in `stform`, as are its first-order derivatives. You can create an `stform` directly from its centers and coefficients using `stmak`. As with all forms, you can evaluate the splines in `stform` with `fnval`, plot them with `fnplt`, etc.

See “Fitting Values at Scattered 2-D Sites” and “The `stform`” in the Spline Toolbox documentation for more information.

splinetool Command History M-File

In the Spline Tool GUI (splinetool), you can now generate a function M-file that you can use to generate, from the original data, any or all graphs currently in the GUI. Select **File** -> **Save M-File** to save the M-file.

This M-file also provides you with a written record of the Spline Toolbox commands used to generate the current graph(s), and can be edited, or executed in a loop over different data sets.

Spline-Specific Zero Finder

A new command `fnzeros` enables you to find an ordered list of the zeros of a univariate, scalar-valued spline, in its basic interval or in a specified interval.

Spline-Specific Minimum Finder

A new command `fnmin` enables you to obtain the minimum or maximum value of a univariate, scalar-valued spline as well as its location, in its basic interval or in a specified interval.

Function Summary

Version 3.1 of the Spline Toolbox provides the following:

- “New Functions” on page -19
- “Functions with New or Changed Capabilities” on page -20

New Functions

Function	Purpose
<code>fnmin</code>	Minimum of a function in a given interval
<code>fnzeros</code>	Find zeros of a function in a given interval
<code>stcol</code>	Scattered translates collocation matrix
<code>stmak</code>	Put together a function in stform
<code>tpaps</code>	Thin-plate smoothing spline

Functions with New or Changed Capabilities

Function	Purpose
fnbrk	A new value of the part argument, 'variables', causes fnbrk to return the number of variables of the specified function.
fnrfn	As a new default, fnrfn now refines the partition (breaks or knots) of the given form by adding to it every midpoint. fnrfn previously had no default.
splinetool	splinetool now permits the entry of a vector whose elements replace that many entries in the weights display, starting with the marked entry.
csaps fnbrk fnplt optknt slvblk spaps spcol spcrv	These functions now treat an empty optional argument the same as not having that optional argument.

Compatibility Consideration

playshow Command Needed to Run Slideshow Style Demos

Starting in R13, to run slideshow style demos from the command line, you must use the playshow command. For example,

```
playshow splxmpl
```

You can continue to run other styles of demos from the command line by typing just the demo name. Spline Toolbox demos that are affected: splxmpl, histodem, ppalldem, spapidem, getcurv2, and spalldem.

Version 3.0 (R12) Spline Toolbox

This table summarizes what's new in Version 3.0 (R12):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details listed under Compatibility Considerations , below. See also Summary	List of bug fixes	No

New features and changes introduced in this version are

- Spline Tool Provides a Visual Interface to the Spline Toolbox
- Automatic Knot Choice Simplifies Use of spapi and spap2
- Automatic Smoothing Parameter Choice Simplifies Use of csaps
- Use of Rational Splines
- B-Spline Visual Interface
- Other New Functions
- Other Enhancements
- Compatibility Considerations
- Bug Fixes

Spline Tool Provides a Visual Interface to the Spline Toolbox

The `splinetool` function invokes a new visual interface that allows you to:

- Import data in various ways
- Try out and compare all the different spline fits available
- Vary the parameters, including the data
- Look at the error

- Look at the derivatives
- View the toolbox functions that produce specific spline fits
- Print the graphs, and save approximations for subsequent use

Automatic Knot Choice Simplifies Use of `spapi` and `spap2`

Until this release, if you wanted to construct a spline interpolant to given data, you had to specify the spline space from which this spline was to be chosen, by providing an appropriate knot sequence. Starting with this release, if you are not so certain about how to choose knots, you can simply specify the order of the spline to be used instead, and `spapi` will provide a suitable knot sequence.

The same difficulty of having to choose a knot sequence occurred in the construction of a least-squares spline approximation to given data, and here, too, you can instead merely specify the number of polynomial pieces of the given order to be used in the approximating spline in `spap2`.

Automatic Smoothing Parameter Choice Simplifies Use of `csaps`

You can now use `csaps` without specifying the smoothing parameter to be used. If none is specified, `csaps` will optionally return the one it chose for the given data, for further experimentation.

Use of Rational Splines

The relevant function functions (e.g., `fnval`, `fnplt`, `fnt1r`, `fnbrk`, `fnrfn`, `fn2fm`, etc.) can now operate on rational splines (NURBS). Specific examples of a rational spline are provided by `rsmak`. Both `rsmak` and `rpmak` are available to generate arbitrary rational splines in B-form and `ppform`, respectively.

B-Spline Visual Interface

Splines in the Spline Toolbox are constructed as a linear combination of B-splines. Run `bspligui` to show how such a B-spline varies as you vary its knots. You can:

- Add knots
- Delete knots

- Move knots or breaks
- Increase/decrease the multiplicity of a knot

Other New Functions

The following functions have been added in the Spline Toolbox 3.0:

- `aptknt(x,k)` provides a good knot sequence for interpolation by splines of order k to data at x .
- `fndir` is available for the construction of directional derivatives, and hence of Jacobians, gradients, and surface normals.
- `fnt1r` is available for the calculation of derivative values; this is particularly useful for rational splines for which formal differentiation is inefficient.
- `chbpnt(knots,k)` provides a good data site sequence for interpolation by splines of order k with knot sequence `knots`.

Other Enhancements

- Both `csaps` and `spaps` can now work with a nonconstant weight in the roughness measure.
- Both `csaps` and `spaps` can also now deal better with near-zero error weights.
- You can now give `spaps` a smoothing parameter rather than a tolerance.
- `fnbrk` can now change the basic interval of any form.
- You can now make `fnval` treat splines as continuous from the left.
- You can also now use `fnval` in the form `fnval(x,f)` as needed for `fzero`, `fminbnd`, etc.
- `fnplt` can now be made not to break the graph of a function at a jump.
- The second argument of `newknt(fn,newl)` has become optional.
- `aveknt(x,k)` can now handle an x of length k (of use in `aptknt`).
- `optknt` can now handle much more nonuniformly spaced data sites, particularly by using `optknt(tau,k,maxiter)` to increase the maximum number of steps used to iteratively solve for the optimal knots.

Compatibility Considerations

optknt and newknt Output No Longer Needs To Be Run Through augknt

In the Spline Toolbox 3.0, you no longer need to run the output from `optknt` and `newknt` through `augknt` to get a complete knot sequence. Running that output through `augknt` now does not cause any problems, but is unnecessary.

Bug Fixes

The Spline Toolbox 3.0 includes several bug fixes, including the following particularly important bug fix.

spmak and ppmak Use Size Argument to Correctly Construct Multivariate Splines

If the spline to be constructed by `spmak` (`knots`, `coefs`) is multivariate, but is meant to be a constant function without any interior knots in its last variable, then the last dimension of `coefs` is necessarily 1.

For example, a one-piece trivariate constant function would have knots stored in a length 3 cell array and a 3-D coefficient array.

```
knots = {[0 1],[0 1],[0 1]};  
coefs = ones([1,1,1]);
```

However, this `coefs` array would be truncated to 2-D because MATLAB suppresses all trailing singleton dimensions (dimensions greater than 2 whose size is 1). Thus in earlier versions of the toolbox, this would fail.

```
spmak(knots, coefs)  
??? Error using ==> spmak  
coefs must be a ([1+]length(KNOTS))-dimensional array
```

Even if you managed to construct the B-form of this constant function, the various `fn...` functions could not work with it.

For the Spline Toolbox 3.0, all these `fn...` functions now handle splines with coefficient arrays of this kind correctly. You can now specify the intended size of the coefficient array by

```
spmak(knots,coefs,sizec)
```

where `sizec` has the intended dimensionality of `coefs`. Now


```
spmak({[0 1],[0 1],[0 1]},ones([1,1,1]),[1 1 1])
```

will correctly construct a one-piece trivariate constant function, as will this.

```
spmak({[0 1],[0 1],[0 1]},1,[1 1 1])
```

Analogously, you can now use the optional third input argument, `d`, in

```
ppmak(breaks, coefs, d)
```

to specify the intended dimensions of the coefficient array.

Compatibility Summary for Spline Toolbox

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 V3.3 (R2006a)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • rsmak Input and Output For Getting an Arc Is Changed • Part 'k...' For a Ppform Disallowed • Strange Error Message from fnval Eliminated
V3.2.2 (R14SP3)	See these Compatibility Considerations subheadings: <ul style="list-style-type: none"> • B-form Evaluation Outside the Basic Interval Returns 0 • spaps Handles Vector-Valued Data Correctly in the Case of Zero Tolerance. • fncmb Returns Error if Only One Coefficient Array Is a Scalar
V3.2.1 (R14SP1)	None
V3.2 (R13+)	See these Compatibility Considerations subheadings: <ul style="list-style-type: none"> • NaNs and Infs • csape Input Simplified • spap2, spapi Input Requirements Updated

V3.1.1 (R13)	See this Compatibility Considerations subheading: <ul style="list-style-type: none">• playshow Command Needed to Run Slideshow Style Demos
V3.0 (R12)	See this Compatibility Considerations subheading: <ul style="list-style-type: none">• optknt and newknt Output No Longer Needs To Be Run Through augknt