Synopsis
fftservertest()
include "fftserver.g"

Description
This class purports to test the fftserver class. In fact, all it does currently is run fftserverdemo.

See Also
fftserver fftserverdemo

Returns
Bool: T for success

1.5.14 interpolate1d - Tool

Package utility
Module mathematics

Various sorts of 1-dimensional interpolation
include "interpolate1d.g"

Constructors

interpolate1d	Construct an interpolate1d `tool`

Functions

done	Delete the interpolate1d tool
initialize	Set the data and the interpolation function
interpolate	Perform an interpolation
setfunction	Change the interpolation function
type	Return the type of this tool

Description

This tool does one-dimensional interpolation on a user supplied set of (x, y) values. The computation is performed in a pre-compiled executable. It is not interpreted by Glish.
`interpolate1d` is used to interpolate between values using any of the

following algorithms:

nearest neighbor This simply returns the y-value for the data point that has the nearest x-value to the requested x-value. You could argue that this is not interpolation at all. But it is fast.

linear Linear interpolation is the simplest useful interpolation scheme. It fits a line between the two data points whose x-value straddle the requested x-value. It is quite a robust interpolation scheme.

cubic polynomial This fits a cubic polynomial to the four points whose x-value straddle the requested x-value.

spline This fits a natural cubic spline to the entire data set and then interpolates on this function. A natural cubic spline is a curve that is smooth in its first derivative, continious in its second and has a zero second second derivative at the end points. This interpolation scheme uses the entire data set to compute the interpolating function unlike all the others described above which only use nearby points.

Demo and test functions are available: `interpolate1ddemo()` and `interpolate1dtest()`.

Example

To use the functions in this tool we have to firstly (1) load the definition of the interpolate1d tool and (2) construct a tool that will actually perform the computations.

```
include "interpolate1d.g"     # 1
interp := interpolate1d()     # 2
```

OK, lets create some fake data to interpolate: $y = 3x^2 + 5$

```
x := 1:10          # 1,2,3,4,5,6,7,8,9,10
y := 3*x*x + 5     # 8, ..., 305
```

Next we need to initialize the interpolation tool — provide it with the x,y data and tell it what interpolation function we want:

```
interp.initialize(x,y,'linear')   # Returns F if it fails
```

We actually did not have to specify `linear`, because that is the default interpolation function. After the interpolator is initialized, we can use it to interpolate the value at some new values of `x`:

```
interp.interpolate([1.5, 2.5])
# [12.5 24.5]
```

Obviously our linear interpolation isn't doing the best possible job with a second order polynomial, so instead we could try a cubic polynomial:

```
interp.setfunction('cubic')
interp.interpolate([1.5, 2.5])
# [11.75 23.75]
```

This is exact of course.

See Also
interpolate1ddemo interpolate1dtest

interpolate1d - Constructor

Package	**utility**
Module	**mathematics**
Tool	**interpolate1d**

Construct an interpolate1d `tool`

Synopsis
interpolate1d()

Description
Use this constructor to construct an interpolate1d tool.

Returns
An interpolate1d tool

Example

```
include 'interpolate1d.g'
interp := interpolate1d();
```

---

interpolate1d.initialize - Function

Package	**utility**
Module	**mathematics**
Tool	**interpolate1d**

Set the data and the interpolation function

Synopsis
initialize(x, y, function)

Arguments

x	independent data variable Allowed:Real valued array.
y	dependent data variable Allowed:Real valued array. Must be the same length as x.
function	interpolation function to use Allowed:'linear', 'nearest', 'cubic', 'spline' Default: 'linear'

Returns
Bool: T if successful

Description
Set the data and the interpolation function.

---

interpolate1d.interpolate - Function

Package	**utility**
Module	**mathematics**
Tool	**interpolate1d**

Perform an interpolation

Synopsis
interpolate(x)

Arguments

x	Perform the interpolation for these positions Allowed:Real valued array

Returns
Real array: interpolated values (same length as x)

Description
Perform the actual interpolation. The results are undefined if initizlize has not been called first.

---

interpolate1d.setfunction - Function

Package	**utility**
Module	**mathematics**
Tool	**interpolate1d**

Change the interpolation function

Synopsis
setfunction(function)

Arguments

function	interpolation function to use Allowed:'linear', 'nearest', 'cubic', 'spline' Default: 'linear'

Description
Change the interpolation function.

---

interpolate1d.done - Function

Package	**utility**
Module	**mathematics**
Tool	**interpolate1d**

Delete the interpolate1d tool

Synopsis
done()

Description
The done function frees up all the memory associated with a interpolate1d tool. After calling this function the interpolate1d tool cannot be used. This function will shut down the server process if no other tools are using the numerics server.

Returns
T

Example

```
include 'interpolate1d.g'
server := interpolate1d();
# ... manipulate the server as you see fit ...
server.done(); # shut down the interpolate1d tool
```

---

interpolate1d.type - Function

Package	**utility**
Module	**mathematics**
Tool	**interpolate1d**

Return the type of this tool

Synopsis
type()

Returns
string

Description

This function returns the string ‘interpolate1d’. It is used so that in a script, you can make sure this *Glish* variable is an interpolate1d `tool`.

1.5.15 interpolate1ddemo - Function

Package utility
Module mathematics

Demonstrate the use of functions in interpolate1d

Synopsis
interpolate1ddemo()

Description
Demonstrate the use of the functions in interpolate1d. Inputs and results are sent to the logger.

See Also
interpolate1d

Returns
Bool: T for success

1.5.16 interpolate1dtest - Function

Package utility
Module mathematics

Test the interpolate1d class

Synopsis
interpolate1dtest()

Description
This class purports to test the interpolate1d class. In fact, all it does currently is run interpolate1ddemo.