**NAME**

SP, EM, GM, TM, TRI, APOD – NMRPipe Window Functions.

**SYNOPSIS**

**nmrPipe –fn SP** [**–off** *offset*] [**–end** *end*] [**–pow** *pow*]

**nmrPipe –fn EM** [**–lb** *lbHz*]

**nmrPipe –fn GM** [**–g1** *invExpHz*] [**–g2** *gaussHz*] [**–g3** *gCenter*]

**nmrPipe –fn GMB** [**–lb** *lbVal*] [**–gb** *gbVal*]

**nmrPipe –fn TM** [**–t1** *leftPts*] [**–t2** *rightPts*]

**nmrPipe –fn TRI** [**–loc** *apexLoc*] [**–lHi** *lHi*] [**–rHi** *rHi*]

**nmrPipe –fn APOD** [**–qName** *apodName*] [**–q1** *apodQ1*] [**–q2** *apodQ2*] [**–q3** *apodQ3*]

**nmrPipe –fn** [**SP | EM | GM | TM | TRI | APOD**] [**–size** *aSize*] [**–start** *aStart*] [**–c** *fScale*] [**–one**] [**–hdr**] [**–inv**]

**DESCRIPTION**

**SP** applies a sine–bell window with an adjustable offset, endpoint, and exponent. The offset and endpoint are specified in units of *pi* radians.

**EM** applies an exponential window. The exponential is specified in terms of a Lorentzian line broadening in Hz.

**GM** applies a Lorentz–to–Gauss window, which is a combination of an inverse exponential and a Gaussian. The purpose of the window is to replace the exponential decay of the original data with a Gaussian decay. The exponential term is specified as a Lorentzian line sharpening in Hz. The Gaussian term is specified as a Gaussian line broadening in Hz, and a center position (location of the Gaussian maximum). The Gaussian center position is specified as a value ranging from 0.0 (Gaussian maximum at the first point of the FID) to 1.0 (Gaussian maximum at the last point of the FID).

**GMB** applies another kind of exponential/Gauss window, inspired by Bruker's Gaussian window function.

**TM** applies a trapezoid window. The window is specified in terms of the lengths of the left and right edges of the trapezoid. The lengths are specified in points.

**TRI** applies a triangle window. The window is specified in terms of the location of the triangle apex, and the heights of the window at the first and last points of the FID.

**APOD** applies one of the windows above, as specified by command–line arguments or header values. It can be used to apply a window function which was specified during format conversion or during previous processing.

In addition to function–specific options, each of the **nmrPipe** window functions provides the following features:

Optional scaling of the first point of each FID.

Automatic recording of window parameters and first point scaling in the data header during processing.

Automatic adjustment of the default size of the window function to match the valid time–domain size

recorded in the header.

Optional extraction and use of window parameters recorded in the header, rather than from the command−line.

An inverse mode, which divides by the window function and first point scale rather than multiplying by them.

Options to adjust the size or starting point of the window function via command−line arguments.

**SP OPTIONS**

**−off** *offset*

(Q1) Specifies the starting point of the sine−bell in units of *pi* radians. Common values are 0.0 (for a sine window which starts height at 0.0) and 0.5 (for a cosine window, which starts at height 1.0). The default value is 0.0.

**−end** *end*

(Q2) Specifies the ending point of the sine−bell in units of *pi* radians. Common values are 1.0 (for a window which goes to 0.0 height at the last point) and 0.95 (for a window which doesn't go all the way to 0.0). The default value is 1.0.

**−pow** *pow*

(Q3) Specifies the exponent of the sine−bell; Non−integer values are allowed. Common values are 1.0 (for ordinary sine−bell) and 2.0 (for squared−bell functions). The default value is 1.0.

**EM OPTIONS**

**−lb** *lbHz*

(Q1) Specifies the exponential decay of the window in terms of a line broadening in Hz. Negative values will generate an increasing exponential window, which corresponds to a line sharpening. The line−broadening parameter is often selected to match the natural linewidth.

**GM OPTIONS**

**−g1** *invExpHz*

(Q1) Specifies the inverse exponential to apply in terms of a line sharpening in Hz. It is usually adjusted to match the natural linewidth. The default value is 0.0, which means no exponential term will be applied, and the window will be a pure Gaussian function.

**−g2** *gaussHz*

(Q2) Specifies the Gaussian to apply in terms of a line broadening in Hz. It is usually adjusted to be somewhat larger (x 1.25 – 4.0) than the line sharpening specified by the **−g1** option.

**−g3** *gCenter*

(Q3) Specifies the position of the Gaussian function's maximum on the FID. It is specified as a value ranging from 0.0 (Gaussian maximum at the first point of the FID) to 1.0 (Gaussian maximum at the last point of the FID). It most applications, the default value of 0.0 is used.

**GMB OPTIONS**

**−lb** *lbVal*

(Q1) Specifies an exponential factor, as used in the formula given below. This value is usually specified as a negative number which is about the same size as the natural linewidth in Hz. The default value is 0.0, which means no exponential term will be applied.

**−gb** *gbVal*

(Q2) Specifies a Gaussian factor, as used in the formula given below. It is usually specified as a positive number which is a fraction of 1.0. The default value is 0.0, which leads to an undefined window function according to the formula; for this reason, the Gaussian term is omitted from the calculation when **−gb 0.0** is given.

**TM OPTIONS**

**−t1** *leftPts*

(Q1) Specifies the number of points in the left edge of the trapezoid window. The window function will increase from 0.0 to 1.0 over this point range.

**−t2** *rightPts*

(Q2) Specifies the number of points in the right edge of the trapezoid window. The window function will decrease from 1.0 to 0.0 over this point range.

**TRI OPTIONS**

**−loc** *apexLoc*

(Q1) Specifies the point location of the triangle function's apex on the FID. The triangle function will have a height of 1.0 at this location.

**−lHi** *lHi*

(Q2) Specifies the height of the triangle function at the first point of the window. The window will vary linearly from this height to 1.0 between the first point and the apex of the window.

**−rHi** *rHi*

(Q3) Specifies the height of the triangle function at the last point of the window. The window will vary linearly from 1.0 to this height between the apex of the window and the last point.

**APOD OPTIONS**

**−qName** *apodName*

Specifies the name of the window function to apply, **SP**, **EM**, etc.

**−q1** *apodQ1*

Specifies the first parameter (Q1) of the selected window function.

**−q2** *apodQ2*

Specifies the second parameter (Q2) of the selected window function.

**−q3** *apodQ3*

Specifies the third parameter (Q3) of the selected window function.

**GENERIC OPTIONS**

**−size** *aSize*

Specifies the number of points in the window function. The default value is the valid time−domain size recorded in the data header.

**−start** *aStart*

Specifies the starting point of the window function. The default value is 1, which means the window function starts at the first point of the FID. This option is intended for creation of composite windows by application of different functions to different regions of the FID.

**−c** *fScale*

Specifies the scaling applied to the first point of the FID, which influences the zero−order offset in the corresponding spectrum. The default value is 1.0, which means no first point adjustment is applied. A value of 0.5 is usually appropriate in cases where no substantial first−order phase correction will be applied.

**−one** This flag influences the values used "outside" the window function, in cases where the window size is smaller than the actual number of data points. By default, data values outside the window region are multiplied by zero when the window is applied. However if the **−one** flag is used, data values outside the window region will be multiplied by 1.0 when the window is applied. This flag is intended to assist creation of composite windows by application of different functions to different regions of the FID.

**−hdr** When this flag is used, default window parameters (Q1,Q2, and Q3) will be extracted from the data header, along with the first point scaling. This requires that *all* of these parameters have already been recorded, for instance during previous processing or format conversion (see

**EXAMPLES** below). Additional command−line can be used to override values restored from the header. The window parameters stored in the data header can be viewed using the **showhdr**(1) program (**showhdr −verb** *dataFileName*).

**−inv** When this flag is used, the inverse (1/window) of the selected window and first point scale will be applied. This option is intended for removing a previously−applied window in inverse processing schemes. This option should generally only be used on window functions which have no values close or equal to zero. In cases where the window does have a zero value, the inverse window is also given as zero.

**SELECTED FORMULAS**

In the following formulas, `tSize` is the number of time−domain points, which defines the length of the window function; `w[i]` is the window function from `i = 0` (first point) to `i = tSize − 1` (last point); `sw` is the sweep width in Hz.

**SP**:

```
w[i] = sin( PI*off  + PI*(end−off)*i/(tSize−1) )^pow
```

**EM**:

```
w[i] = exp( −PI*i*lb/sw )
```

**GM**:

`w[i] = exp( e − g*g )`, where

```
e = PI*i*g1/sw
g = 0.6*PI*g2*(g3*(tSize−1) − i)/sw
```

**GMB**:

`w[i] = exp( −a*t − b*t*t )`, where

```
t  = i/sw
aq = tSize/sw
a  = PI*lb
b  = −a/(2.0*gb*aq)
```

**EXAMPLES**

Two ways of specifying an ordinary cosine bell:

```
nmrPipe −fn SP −off 0.5
nmrPipe −fn SP −off 0.5 −end 1.0
```

Three ways of specifying an ordinary sine bell; note that this is the default function for SP:

```
nmrPipe −fn SP
nmrPipe −fn SP −off 0.0
nmrPipe −fn SP −off 0.0 −end 1.0
```

A cosine−squared bell:

```
nmrPipe −fn SP −off 0.5 −end 1.0 −pow 2
```

A 60−degree shifted sine bell with scaling of the first point by 0.5; offset = 60/180, roughly 0.33:

```
nmrPipe −fn SP −off 0.33 −end 1.0
```

A cosine bell which does not decrease all the way to zero; this window function can usually be inverted safely for inverse processing schemes, because its smallest height is about 0.16:

```
nmrPipe -fn SP -off 0.5 -end 0.95
```

A cosine-squared roll-off function; the cosine function is set to span 100 points starting from point 257 in the data, so that the window region extends from point 257 to point 356. Since the **-one** flag is included, the data will be multiplied by 1.0 outside of this region. Therefore, the result is a window which is uniformly 1.0 over points 1 to 256, and decays to 0.0 as a cosine-square over points 257 to 356:

```
nmrPipe -fn SP -off 0.5 -pow 2 -start 257 -size 100 -one
```

A typical usage of a Lorentz-to-Gauss window:

```
nmrPipe -fn GM -g1 20 -g2 25
```

The following scheme shows window parameters (APOD, Q1, Q2, and Q3), first point scale (C1), and phasing (P0, P1) specified during conversion. The values are then extracted and used during processing by including the **-hdr** option with processing functions **APOD** and **PS**:

```
#!/bin/csh

bruk2pipe -in hsqcn.ser \
 -xN           2048    -yN         256   \
 -xT           1024    -yT         128   \
 -xMODE     Complex    -yMODE  Complex   \
 -xSW       9090.91    -ySW    2500.00   \
 -xOBS      600.138    -yOBS   60.8108   \
 -xCAR         4.73    -yCAR     118.0   \
 -xLAB           HN    -yLAB         N   \
 -xAPOD          SP    -yAPOD       SP   \
 -xQ1          0.50    -yQ1       0.50   \
 -xQ2          0.98    -yQ2       0.95   \
 -xQ3           2.0    -yQ3        1.0   \
 -xC1           0.5    -yC1        1.0   \
 -xP0           0.0    -yP0      -90.0   \
 -xP1           0.0    -yP1      180.0   \
 -ndim            2    -aq2D    States   \
 -out hsqcn.fid -verb -ov



nmrPipe -in hsqcn.fid \
| nmrPipe  -fn SOL                        \
| nmrPipe  -fn APOD -hdr                  \
| nmrPipe  -fn ZF -auto                   \
| nmrPipe  -fn FT                         \
| nmrPipe  -fn PS -p0 22 -p1 0.0 -di      \
| nmrPipe  -fn EXT -left -sw -verb        \
| nmrPipe  -fn TP                         \
| nmrPipe  -fn APOD -hdr                  \
| nmrPipe  -fn ZF -auto                   \
| nmrPipe  -fn FT                         \
| nmrPipe  -fn PS -hdr -di                \
   -verb -ov -out test.ft2
```

In this inverse processing scheme, a spectrum is inverse transformed, and the SP window applied in a previous scheme is removed (**SP –inv –hdr**) in order to apply Linear Prediction (**LP**). After **LP**, the window is re–applied (**SP –hdr**):

```
xyz2pipe –in lp/test%03d.ft3 –z –verb              \
| nmrPipe  –fn HT  –auto                           \
| nmrPipe  –fn PS  –inv –hdr                       \
| nmrPipe  –fn FT  –inv                            \
| nmrPipe  –fn ZF  –inv                            \
| nmrPipe  –fn SP  –inv –hdr                       \
| nmrPipe  –fn LP  –ps0–0                          \
| nmrPipe  –fn SP  –hdr                            \
| nmrPipe  –fn ZF  –auto                           \
| nmrPipe  –fn FT                                  \
| nmrPipe  –fn PS  –hdr –di                        \
| pipe2xyz –out lp/test%03d.ft3 –z –inPlace
```

**HEADER VALUES**

The **nmrPipe** window functions use the recorded time–domain size (NDAPOD) to establish their default length.

When the **–hdr** flag is used, default window parameters are extracted from header values NDAPODCODE, NDAPODQ1, NDAPODQ2, NDAPODQ3, and NDC1.

The header values NDAPODCODE, NDAPODQ1, NDAPODQ2, NDAPODQ3, and NDC1 are updated according to the values applied during processing.

**SEE ALSO**

**nmrPipe**(1), **xyz2pipe**(1), **fdatap**(1),