

Documentation for BasisFns.h and BasisFns.c

Steven Andrews, © 2003

See the document “LibDoc” for general information about this and other libraries.

```
int getbasis(set bset);
float spectbasis(float x,float *param,sptr spec,float *deriv);
float constbasis(float x,float *param,sptr spec,float *deriv);
float linebasis(float x,float *param,sptr spec,float *deriv);
float expbasis(float x,float *param,sptr spec,float *deriv);
float logbasis(float x,float *param,sptr spec,float *deriv);
float quadbasis(float x,float *param,sptr spec,float *deriv);
float asinbasis(float x,float *param,sptr spec,float *deriv);
float gaussbasis(float x,float *param,sptr spec,float *deriv);
float xgaussbasis(float x,float *param,sptr spec,float *deriv);
float sinbasis(float x,float *param,sptr spec,float *deriv);
float lorentzbasis(float x,float *param,sptr spec,float *deriv);
float peakbasis(float x,float *param,sptr spec,float *deriv);
float peak1basis(float x,float *param,sptr spec,float *deriv);
float peak2basis(float x,float *param,sptr spec,float *deriv);
float peakzbasis(float x,float *param,sptr spec,float *deriv);
float diffusebasis(float x,float *param,sptr spec,float *deriv);
float diffuse2basis(float x,float *param,sptr spec,float *deriv);
float convexpbasis(float x,float *param,sptr spec,float *deriv);
float rationbasis(float x,float *param,sptr spec,float *deriv);
```

Requires: <string.h>, <float.h>, <math.h>, <stdio.h>, "math2.h", "Set.h",
"Spectra.h", "BasisFn.h", "BasisFns.h", "Rn.h", "RnSort.h", "string2.h"

Example program: SpectFit.c

Moved from BasisFn.c library 3/10/02.

This library is the code for the basis functions used by the library BasisFn.c and the program *SpectFit*. Basis functions are basically simple functions that are used in combination for fitting data. Each basis function needs some structure members to be defined, such as parameter names and default values, and it needs a function to be written to calculate the basis function value in terms of the independent variable, x , and the function parameters. For fitting and error estimation, the function also needs to be able to return the basis function derivatives in terms of the parameters.

To add a new basis function, it needs to be added in four ways: 1) write a routine for it. This routine is sent an x value, a list of parameters (such as weighting, peak location, and peak width), and an optional constant spectrum; it should return the y value and, if necessary, all derivatives with respect to parameters. 2) Declare the new routine in the header file, BasisFns.h. 3) Add the function to the routine *getbasis*, so *SpectFit* knows it exists. Previously defined basis functions are useful guides for what's required. 4) Add it to both this documentation and to the *SpectFit* documentation.

DeclareBasis is a local function, not declared in the header file, which simplifies the process of adding new basis functions to the set *bset*. The name, description, address, list of default parameter values, and list of parameter names are sent it, along with a pointer to the set. The basis function is set up and added to the set.

Either a pointer to the newly created function is returned, or NULL if memory could not be allocated.

getbasis fills an empty set (basisfns) with the structures of all available basis functions, with the parameters set to default values.

constant	constbasis	A simple constant offset.
$y=offset$		
offset	1	The amount of offset.
spectrum	spectbasis	The value of a spectrum, interpolated as necessary.
$y=weight*spectrum(x)$		
weight	1	Weighting factor.
line	linebasis	A straight line through $(x0,0)$.
$y=slope*(x-x0)$		
slope	0.001	The slope of the line.
$x0$	0	x position where the line crosses the x -axis.
exp	expbasis	Exponential function.
$y=factor*exp(slope*x)$		
factor	1	The pre-exponential factor.
slope	0.1	Exponential slope.
log	logbasis	Natural log function.
$y=weight*\ln(slope*x+intercept)$, or 0 if argument is ≤ 0		
weight	10^{-4}	Function weight.
slope	1	Slope of argument.
intercept	1	Intercept of argument.
quad	quadbasis	A quadratic in standard format.
$y=curve*(x-x0)^2+slope*(x-x0)+intercept$		
curve	10^{-6}	Curvature.
slope	10^{-5}	Slope.
intercept	0.01	y -intercept.
$x0$	0	x shift.
asinh	asinbasis	Inverse hyperbolic sine function.
$y=weight*asinh(slope*x+intercept)$		
weight	10^{-4}	Function weight.
slope	1	Slope of argument.
intercept	1	Intercept of argument.
gaussian	guassbasis	A standard Gaussian.
$y=area*(std_dev*\sqrt{2\pi})*exp[-(x-mean)^2/(2*std_dev^2)]$, or 0 if std_dev is 0		
area	1	Total area of Gaussian.
mean	0	Mean of Gaussian.
std_dev	1	Standard deviation of Gaussian, ≥ 0 .

xgauss xgausbasis Gaussian times x ; useful for heterogeneously broadened spectral lines.
 $y=x*area/(std_dev*\sqrt{2\pi})*\exp[-(x-mean)^2/(2*std_dev^2)]$, or 0 if std_dev is 0
area 0.0002 Similar, but not equal, to the area.
mean 1945 Close to the mean.
std_dev 4 Close to the standard deviation, ≥ 0 .

sine sinbasis Sine wave.
 $y=amp*\sin(freq*x+shift)$
amp 1 Amplitude, baseline to peak.
freq 1 Frequency, in radian units.
shift 0 Phase shift, in radians.

lorentz lorentzbasis A standard Lorentzian.
 $y=max/\{1+[(x-mean)/(fwhm/2)]^2\}$
max 1 Peak height.
mean 0 Peak center.
fwhm 1 Full width at half maximum, ≥ 0 .

peak peakbasis An x -weighted sum of a Gaussian and a Lorentzian. Useful for spectroscopy, with homogeneously and heterogeneously broadened lines.
 $y=x/position*[(1-shape)*gauss(x-position)+shape*lorentz(x-position)]$
 $gauss(x)=\exp(-4*\ln(2)*x^2/width^2)$
 $lorentz(x)=1/(1+4*x^2/width^2)$
height 1 Maximum peak height.
position 2250 Peak center, ignoring skewing.
fwhm 10 Full width at half maximum, ≥ 0 .
shape 0.5 Fraction of height that is from lorentzian, 0 to 1.

peakd1 peak1basis First derivative of a peak function, using x -weighted differentiation. Useful for Stark effect fitting.
 $y=x*\{\partial/\partial x [peak(x)/x]\}$
peak(x) is defined by the “peak” basis function
height 1 Maximum of the peak that is differentiated.
position 2250 Center of the peak that is differentiated.
fwhm 10 FWHM of the peak that is differentiated, ≥ 0 .
shape 0.5 Shape of the peak that is differentiated, 0 to 1.

peakd2 peak2basis Second derivative of a peak function, using x -weighted differentiation. Useful for Stark effect fitting.
 $y=x*\{\partial^2/\partial x^2 [peak(x)/x]\}$
peak(x) is defined by the “peak” basis function
height 1 Maximum of the peak that is differentiated.
position 2250 Center of the peak that is differentiated.
fwhm 10 FWHM of the peak that is differentiated, ≥ 0 .

<i>shape</i>	0.5	Shape of the peak that is differentiated, 0 to 1.
peakz	peakzbasis	Sum of zeroth, first and second derivatives of a peak function. Useful for Stark effect fitting. $y=z0*peak(x)+z1*x*\{\partial/\partial x [peak(x)/x]\}+z2*x*\{\partial^2/\partial x^2 [peak(x)/x]\}$ peak(x) is defined by the "peak" basis function
<i>z0</i>	0.0001	Zeroth derivative contribution.
<i>z1</i>	0.001	First derivative contribution.
<i>z2</i>	0.01	Second derivative contribution.
<i>height</i>	1	Maximum peak height.
<i>position</i>	2250	Peak center, ignoring skewing.
<i>fwhm</i>	10	Full width at half maximum.
<i>shape</i>	0.5	Fraction of the height that is lorentzian contribution.
diffuse	diffusebasis	A standard gaussian multiplied by a spectrum. Useful for finding diffusion values, using Fourier transforms of the concentrations at two times. $y=spectrum(x)*area*\exp(-dt*x^2)$
<i>area</i>	1	Area of Fourier transformed Gaussian.
<i>dt</i>	1	Diffusion constant times time.
diffuse2	diffuse2basis	A squared error function. Was used for diffusion out of a square 2-D box. $y=c0*erf^2\{\sqrt{[td/(x-t0)]}/4\}$, or <i>c0</i> if $x \leq t0$
<i>c0</i>	100	Initial value.
<i>td</i>	1	Diffusion time, equal to box area/diffusion const.
<i>t0</i>	0	Time diffusion starts.
decay	convexpbasis	Convolution of a gaussian with an exponential that turns on at $x=0$. Useful for pump-probe spectroscopy, where the pump beam autocorrelation is the Gaussian and the time response is the exponential. $y=convolution\ of\ \{height*\exp(-kt),\ or\ 0\ if\ t < 0\}\ with\ 1/[\sigma\sqrt{(2\pi)}]*\exp[-t^2/(2\sigma^2)]$ $=height/2*\exp(-kt+\sigma^2 k^2/2)*\{1+erf[t-\sigma^2 k/(\sigma\sqrt{2})]\}$ $k=1/tau$, and is decay rate $\sigma=fwhm/[2\sqrt{(2\ln 2)}]$, and is standard deviation of autocorrelation $t=x-shift$, and is time since start of exponential
<i>height</i>	1	Initial height of exponential.
<i>fwhm</i>	0.16	FWHM of gaussian, with unit area.
<i>tau</i>	1	Time constant of exponential
<i>shift</i>	10	<i>x</i> value where exponential turns on.
rational	rationbasis	A rational function, which fits almost anything. Note that there is redundancy in the equation, so some parameters should be fixed. $y=(n0+n1*x+n2*x^2+n3*x^3+n4*x^4+n5*x^5)/(d0+d1*x+d2*x^2+d3*x^3+d4*x^4+d5*x^5)$
<i>n0</i>	1	Numerator constant coefficient.
<i>n1</i>	0	Numerator linear coefficient.
<i>n2</i>	0	Numerator quadratic coefficient.

<i>n3</i>	0	Numerator cubic coefficient.
<i>n4</i>	0	Numerator quartic coefficient.
<i>n5</i>	0	Numerator quintic coefficient.
<i>d0</i>	1	Denominator constant coefficient.
<i>d1</i>	0	Denominator linear coefficient.
<i>d2</i>	0	Denominator quadratic coefficient.
<i>d3</i>	0	Denominator cubic coefficient.
<i>d4</i>	0	Denominator quartic coefficient.
<i>d5</i>	0	Denominator quintic coefficient.