For backward compatibility, if you don't include *accuracy*, gammp uses older code that is slower for an equivalent accuracy, and cannot achieve as high accuracy.

The ability of gammp to return a value having full fractional accuracy is limited by double-precision calculations. This means that it will mostly have fractional accuracy better than about 10^{-15}, but this is not guaranteed, especially for extreme values of *a* and *x*.

See Also

The **gammaInc** and **gammq** functions.

gammq

`gammq(a, x [, accuracy])`

The gammq function returns the regularized incomplete gamma function 1-P(a,x), where $a > 0$, $x \geq 0$. Optionally, *accuracy* can be used to specify the desired fractional accuracy. Same as `gammaInc(a, x)/gamma(a)`.

Details

The *accuracy* parameter specifies the fractional accuracy that you desire. That is, if you set *accuracy* to 10^{-7}, that means that you wish that the absolute value of $(f_{actual} - f_{returned})/f_{actual}$ be less than 10^{-7}.

For backward compatibility, if you don't include *accuracy*, gammq uses older code that is slower for an equivalent accuracy, and cannot achieve as high accuracy.

The ability of gammq to return a value having full fractional accuracy is limited by double-precision calculations. This means that it will mostly have fractional accuracy better than about 10^{-15}, but this is not guaranteed, especially for extreme values of *a* and *x*.

See Also

The **gammaInc** and **gammp** functions.

Gauss

`Gauss(x,xc,wx [,y,yc,wy [,z,zc,wz [,t,tc,wt]]])`

The Gauss function returns a normalized Gaussian for the specified dimension.

$$Gauss(\mathbf{r},\mathbf{c},\mathbf{w}) = \prod_{i=1}^{n} \frac{1}{w_i\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{r_i - c_i}{w_i}\right)^2\right],$$

where n is the number of dimensions.

Parameters

xc, *yc*, *zc*, and *tc* are the centers of the Gaussian in the X, Y, Z, and T directions, respectively.

wx, *wy*, *wz*, and *wt* are the widths of the Gaussian in the X, Y, Z, and T directions, respectively.

Note that w_i here is the standard deviation of the Gaussian. This is different from the width parameter in the gauss curve fitting function, which is sqrt(2) times the standard deviation.

Note also that the Gauss function lacks the cross-correlation parameter that is included in the Gauss2D curve fitting function.

Examples

```
Make/N=100 eee=gauss(x,50,10)
Print area(eee,-inf,inf)
  0.999999

Make/N=(100,100) ddd=gauss(x,50,10,y,50,15)
Print area(ddd,-inf,inf)
  0.999137
```

See Also

Gauss1D (duplicates the Gauss built-in curve fitting function)

Gauss2D (duplicates the Gauss2D built-in curve fitting function)