(../index.html)

Scipy.org (http://scipy.org/) Docs (http://docs.scipy.org/)

NumPy v1.13 Manual (../index.html) NumPy Reference (index.html)

Routines (routines.html)

index (../genindex.html) next (generated/numpy.sin.html)

previous (generated/numpy.not_equal.html)

Mathematical functions

Trigonometric functions

`sin` (generated/numpy.sin.html#numpy.sin)(x, /[, out, where, casting, order, ...])

Trigonometric sine, element-wise.

`cos` (generated/numpy.cos.html#numpy.cos)(x, /[, out, where, casting, order, ...])

Cosine element-wise.

`tan` (generated/numpy.tan.html#numpy.tan)(x, /[, out, where, casting, order, ...])

Compute tangent element-wise.

`arcsin` (generated/numpy.arcsin.html#numpy.arcsin) (x, /[, out, where, casting, order, ...])

Inverse sine, element-wise.

`arccos` (generated/numpy.arccos.html#numpy.arccos) (x, /[, out, where, casting, order, ...])

Trigonometric inverse cosine, element-wise.

`arctan` (generated/numpy.arctan.html#numpy.arctan) (x, /[, out, where, casting, order, ...])

Trigonometric inverse tangent, element-wise.

`hypot` (generated/numpy.hypot.html#numpy.hypot) (x1, x2, /[, out, where, casting, ...])

Given the “legs” of a right triangle, return its hypotenuse.

`arctan2` (generated/numpy.arctan2.html#numpy.arctan2) (x1, x2, /[, out, where, casting, ...])

Element-wise arc tangent

Table Of Contents (../contents.htm

Previous topic

numpy.not_equal (generated/numpy.n

Next topic

numpy.sin (generated/numpy.s

	of `x1/x2` choosing the quadrant correctly.
`degrees` (generated/numpy.degrees.html#numpy.degrees)(x, /[, out, where, casting, order, ...])	Convert angles from radians to degrees.
`radians` (generated/numpy.radians.html#numpy.radians)(x, /[, out, where, casting, order, ...])	Convert angles from degrees to radians.
`unwrap` (generated/numpy.unwrap.html#numpy.unwrap)(p[, discont, axis])	Unwrap by changing deltas between values to 2*pi complement.
`deg2rad` (generated/numpy.deg2rad.html#numpy.deg2rad)(x, /[, out, where, casting, order, ...])	Convert angles from degrees to radians.
`rad2deg` (generated/numpy.rad2deg.html#numpy.rad2deg)(x, /[, out, where, casting, order, ...])	Convert angles from radians to degrees.

Hyperbolic functions

`sinh` (generated/numpy.sinh.html#numpy.sinh)(x, /[, out, where, casting, order, ...])	Hyperbolic sine, element-wise.
`cosh` (generated/numpy.cosh.html#numpy.cosh)(x, /[, out, where, casting, order, ...])	Hyperbolic cosine, element-wise.
`tanh` (generated/numpy.tanh.html#numpy.tanh)(x, /[, out, where, casting, order, ...])	Compute hyperbolic tangent element-wise.
`arcsinh` (generated/numpy.arcsinh.html#numpy.arcsinh)(x, /[, out, where, casting, order, ...])	Inverse hyperbolic sine element-wise.
`arccosh` (generated/numpy.arccosh.html#numpy.arccosh)	Inverse

(x, /[, out, where, casting, order, ...])	hyperbolic cosine, element-wise.
`arctanh` (generated/numpy.arctanh.html#numpy.arctanh)(x, /[, out, where, casting, order, ...])	Inverse hyperbolic tangent element-wise.

Rounding

`around` (generated/numpy.around.html#numpy.around)(a[, decimals, out])	Evenly round to the given number of decimals.
`round_` (generated/numpy.round_.html#numpy.round_)(a[, decimals, out])	Round an array to the given number of decimals.
`rint` (generated/numpy.rint.html#numpy.rint)(x, /[, out, where, casting, order, ...])	Round elements of the array to the nearest integer.
`fix` (generated/numpy.fix.html#numpy.fix)(x[, out])	Round to nearest integer towards zero.
`floor` (generated/numpy.floor.html#numpy.floor)(x, /[, out, where, casting, order, ...])	Return the floor of the input, element-wise.
`ceil` (generated/numpy.ceil.html#numpy.ceil)(x, /[, out, where, casting, order, ...])	Return the ceiling of the input, element-wise.

`trunc` (generated/numpy.trunc.html#numpy.trunc)(x, /[, out, where, casting, order, ...])	Return the truncated value of the input, element-wise.

Sums, products, differences

`prod` (generated/numpy.prod.html#numpy.prod)(a[, axis, dtype, out, keepdims])	Return the product of array elements over a given axis.
`sum` (generated/numpy.sum.html#numpy.sum)(a[, axis, dtype, out, keepdims])	Sum of array elements over a given axis.
`nanprod` (generated/numpy.nanprod.html#numpy.nanprod)(a[, axis, dtype, out, keepdims])	Return the product of array elements over a given axis treating Not a Numbers (NaNs) as ones.
`nansum` (generated/numpy.nansum.html#numpy.nansum)(a[, axis, dtype, out, keepdims])	Return the sum of array elements over a given axis treating Not a Numbers (NaNs) as zero.
`cumprod` (generated/numpy.cumprod.html#numpy.cumprod)(a[, axis, dtype, out])	Return the cumulative product of elements along a given axis.
`cumsum` (generated/numpy.cumsum.html#numpy.cumsum)(a[, axis, dtype, out])	Return the cumulative

	sum of the elements along a given axis.
`nancumprod` (generated/numpy.nancumprod.html#numpy.nancumprod)(a[, axis, dtype, out])	Return the cumulative product of array elements over a given axis treating Not a Numbers (NaNs) as one.
`nancumsum` (generated/numpy.nancumsum.html#numpy.nancumsum)(a[, axis, dtype, out])	Return the cumulative sum of array elements over a given axis treating Not a Numbers (NaNs) as zero.
`diff` (generated/numpy.diff.html#numpy.diff)(a[, n, axis])	Calculate the n-th discrete difference along given axis.
`ediff1d` (generated/numpy.ediff1d.html#numpy.ediff1d)(ary[, to_end, to_begin])	The differences between consecutive elements of an array.
`gradient` (generated/numpy.gradient.html#numpy.gradient)(f, *varargs, **kwargs)	Return the gradient of an N-dimensional array.
`cross` (generated/numpy.cross.html#numpy.cross)(a, b[, axisa, axisb, axisc, axis])	Return the cross product of two (arrays of) vectors.

`trapz` (generated/numpy.trapz.html#numpy.trapz)(y[, x, dx, axis])	Integrate along the given axis using the composite trapezoidal rule.

Exponents and logarithms

`exp` (generated/numpy.exp.html#numpy.exp)(x, /[, out, where, casting, order, ...])	Calculate the exponential of all elements in the input array.
`expm1` (generated/numpy.expm1.html#numpy.expm1)(x, /[, out, where, casting, order, ...])	Calculate `exp(x) - 1` for all elements in the array.
`exp2` (generated/numpy.exp2.html#numpy.exp2)(x, /[, out, where, casting, order, ...])	Calculate _2**p_ for all _p_ in the input array.
`log` (generated/numpy.log.html#numpy.log)(x, /[, out, where, casting, order, ...])	Natural logarithm, element-wise.
`log10` (generated/numpy.log10.html#numpy.log10)(x, /[, out, where, casting, order, ...])	Return the base 10 logarithm of the input array, element-wise.
`log2` (generated/numpy.log2.html#numpy.log2)(x, /[, out, where, casting, order, ...])	Base-2 logarithm of _x_.
`log1p` (generated/numpy.log1p.html#numpy.log1p)(x, /[, out, where, casting, order, ...])	Return the natural logarithm of one plus the input array, element-wise.
`logaddexp` (generated/numpy.logaddexp.html#numpy.logaddexp)(x1, x2, /[, out, where, casting, ...])	Logarithm of the sum of exponentiations of the inputs.
`logaddexp2` (generated/numpy.logaddexp2.html#numpy.logaddexp2)(x1, x2, /[, out, where, casting, ...])	Logarithm of the sum of exponentiations of the inputs in base-2.

Other special functions

`i0` (generated/numpy.i0.html#numpy.i0)(x)	Modified Bessel function of the first kind, order 0.
`sinc` (generated/numpy.sinc.html#numpy.sinc)(x)	Return the sinc function.

Floating point routines

`signbit` (generated/numpy.signbit.html#numpy.signbit)(x, /[, out, where, casting, order, ...])	Returns element-wise True where signbit is set (less than zero).
`copysign` (generated/numpy.copysign.html#numpy.copysign)(x1, x2, /[, out, where, casting, ...])	Change the sign of x1 to that of x2, element-wise.
`frexp` (generated/numpy.frexp.html#numpy.frexp)(x[, out1, out2], / [[, out, where, ...])	Decompose the elements of x into mantissa and twos exponent.
`ldexp` (generated/numpy.ldexp.html#numpy.ldexp)(x1, x2, /[, out, where, casting, ...])	Returns x1 * 2**x2, element-wise.
`nextafter` (generated/numpy.nextafter.html#numpy.nextafter)(x1, x2, /[, out, where, casting, ...])	Return the next floating-point value after x1 towards x2, element-wise.
`spacing` (generated/numpy.spacing.html#numpy.spacing)(x, /[, out, where, casting, order, ...])	Return the distance between x and the nearest adjacent number.

Arithmetic operations

`add` (generated/numpy.add.html#numpy.add) (x1, x2, /[, out, where, casting, order, ...])	Add arguments element-wise.
`reciprocal` (generated/numpy.reciprocal.html#numpy.reciprocal) (x, /[, out, where, casting, ...])	Return the reciprocal of the argument, element-wise.
`negative` (generated/numpy.negative.html#numpy.negative) (x, /[, out, where, casting, order, ...])	Numerical negative, element-wise.
`multiply` (generated/numpy.multiply.html#numpy.multiply) (x1, x2, /[, out, where, casting, ...])	Multiply arguments element-wise.
`divide` (generated/numpy.divide.html#numpy.divide) (x1, x2, /[, out, where, casting, ...])	Divide arguments element-wise.
`power` (generated/numpy.power.html#numpy.power) (x1, x2, /[, out, where, casting, ...])	First array elements raised to powers from second array, element-wise.
`subtract` (generated/numpy.subtract.html#numpy.subtract) (x1, x2, /[, out, where, casting, ...])	Subtract arguments, element-wise.
`true_divide` (generated/numpy.true_divide.html#numpy.true_divide) (x1, x2, /[, out, where, ...])	Returns a true division of the inputs, element-wise.
`floor_divide` (generated/numpy.floor_divide.html#numpy.floor_divide) (x1, x2, /[, out, where, ...])	Return the largest integer smaller or equal to the division of the inputs.
`float_power` (generated/numpy.float_power.html#numpy.float_power) (x1, x2, /[, out, where, ...])	First array elements raised to powers from second array, element-wise.
`fmod` (generated/numpy.fmod.html#numpy.fmod)(x1, x2, /[, out, where, casting, ...])	Return the element-wise remainder of division.
`mod` (generated/numpy.mod.html#numpy.mod) (x1, x2, /[, out, where, casting, order, ...])	Return element-wise

	remainder of division.
`modf` (generated/numpy.modf.html#numpy.modf)(x[, out1, out2], / [[, out, where, ...])	Return the fractional and integral parts of an array, element-wise.
`remainder` (generated/numpy.remainder.html#numpy.remainder)(x1, x2, /[, out, where, casting, ...])	Return element-wise remainder of division.
`divmod` (generated/numpy.divmod.html#numpy.divmod)(x1, x2[, out1, out2], / [[, out, ...])	Return element-wise quotient and remainder simultaneously.

Handling complex numbers

`angle` (generated/numpy.angle.html#numpy.angle)(z[, deg])	Return the angle of the complex argument.
`real` (generated/numpy.real.html#numpy.real)(val)	Return the real part of the complex argument.
`imag` (generated/numpy.imag.html#numpy.imag)(val)	Return the imaginary part of the complex argument.
`conj` (generated/numpy.conj.html#numpy.conj)(x, /[, out, where, casting, order, ...])	Return the complex conjugate, element-wise.

Miscellaneous

`convolve` (generated/numpy.convolve.html#numpy.convolve)(a, v[, mode])	Returns the discrete, linear convolution of two one-

	dimensional sequences.
`clip` (generated/numpy.clip.html#numpy.clip)(a, a_min, a_max[, out])	Clip (limit) the values in an array.
`sqrt` (generated/numpy.sqrt.html#numpy.sqrt)(x, /[, out, where, casting, order, ...])	Return the positive square-root of an array, element-wise.
`cbrt` (generated/numpy.cbrt.html#numpy.cbrt)(x, /[, out, where, casting, order, ...])	Return the cube-root of an array, element-wise.
`square` (generated/numpy.square.html#numpy.square) (x, /[, out, where, casting, order, ...])	Return the element-wise square of the input.
`absolute` (generated/numpy.absolute.html#numpy.absolute) (x, /[, out, where, casting, order, ...])	Calculate the absolute value element-wise.
`fabs` (generated/numpy.fabs.html#numpy.fabs)(x, /[, out, where, casting, order, ...])	Compute the absolute values element-wise.
`sign` (generated/numpy.sign.html#numpy.sign)(x, /[, out, where, casting, order, ...])	Returns an element-wise indication of the sign of a number.
`heaviside` (generated/numpy.heaviside.html#numpy.heaviside) (x1, x2, /[, out, where, casting, ...])	Compute the Heaviside step function.
`maximum` (generated/numpy.maximum.html#numpy.maximum) (x1, x2, /[, out, where, casting, ...])	Element-wise maximum of array elements.
`minimum` (generated/numpy.minimum.html#numpy.minimum) (x1, x2, /[, out, where, casting, ...])	Element-wise minimum of array

	elements.
`fmax` (generated/numpy.fmax.html#numpy.fmax)(x1, x2, /[, out, where, casting, ...])	Element-wise maximum of array elements.
`fmin` (generated/numpy.fmin.html#numpy.fmin)(x1, x2, /[, out, where, casting, ...])	Element-wise minimum of array elements.
`nan_to_num` (generated/numpy.nan_to_num.html#numpy.nan_to_num)(x[, copy])	Replace nan with zero and inf with finite numbers.
`real_if_close` (generated/numpy.real_if_close.html#numpy.real_if_close)(a[, tol])	If complex input returns a real array if complex parts are close to zero.
`interp` (generated/numpy.interp.html#numpy.interp)(x, xp, fp[, left, right, period])	One-dimensional linear interpolation.