

Math-c Documentation

Complex numbers and rounding functions

The complex number is represented by a digit number followed by the letter "i", example "1+3i", "1i", "-5i", etc. If the letter is not with a number, then is considered as a variable and not a complex number, example "1+i", "5*i", etc. Other functions also can use complex numbers as inputs like the trigonometric, power functions etc.

Complex numbers

y = abs(x)

magnitude

x -> value

returns

y -> $\sqrt{x_r^2 + x_i^2}$

y = arg(x)

angle between real and imaginary component

x -> value

returns

y -> $\text{atan2}(x_i, x_r)$, angle in radians

y = cj(x)

conjugate for each element of x

x -> value

returns

y -> conjugate

$y = \text{imag}(x)$

imaginary part of x

x -> value

returns

y -> imaginary part as real number

$y = \text{rect}(m, \phi)$

convert the polar real numbers to a rectangular complex number

m -> real value, magnitude

ϕ -> real value, angle in radians

returns

y -> imaginary number with rectangular coordinate.

$y = \text{unitV}(x)$

return a scalar or quaternion of magnitude 1

x -> scalar value or quaternion

returns

y -> a scalar or quaternion with magnitude 1

Rounding functions

The function is applied for real part and then for imaginary part of number.

$y = \text{ceil}(x)$

Round up number

x -> value

returns

y -> round to nearest integer greater than x, example:

```
>>> ceil([-5.1 -4.9 2.3 6.8+5.1i])
```

```
ans = [-5 -4 3 7+6i]
```

y = floor(x)

Round down number

x -> value

returns

y -> round to nearest integer lower than x, example:

```
>>> floor([-5.1 -4.9 2.3 6.8+5.1i])
```

```
ans = [-6 -5 2 6+5i]
```

y = round(x)

Round nearest number

x -> value

returns

y -> round to nearest integer, example:

```
>>> round([-5.1 -4.9 2.3 6.8+5.1i])
```

```
ans = [-5 -5 2 7+5i]
```

y = trunc(x)

Truncate the decimals of x

x -> value

returns

y -> truncate te decimals, example:

```
>>> trunc([-5.1 -4.9 2.3 6.8+5.1i])
```

```
ans = [-5 -4 2 6+5i]
```