$$centrMassXY = \frac{\sum waveA[i]\ \ waveB[i]}{\sum waveB[i]}.$$

See Also
centerOfMass, **mean**, **areaXY**, **SumDimension**, **ImageAnalyzeParticles**

cequal

cequal(*z1*, *z2*)

The cequal function determines the equality of two complex numbers *z1* and *z2*. It returns 1 if they are equal, or 0 if not.

This is in contrast to the == operator, which compares only the real components of *z1* and *z2*, ignoring the imaginary components.

Examples

```
Function TestComplexEqualities()
    Variable/C z1= cmplx(1,2), z2= cmplx(1,-2)
    // This test compares only the real parts of z1 and z2:
    if( z1 == z2 )
        Print "== match"
    else
        Print "no == match"
    endif
    // This test compares both real and imaginary parts of z1 and z2:
    if( cequal(z1,z2) )
        Print "cequal match"
    else
        Print "no cequal match"
    endif
End
```

```
•TestComplexEqualities()
  == match
  no cequal match
```

See Also
The **imag**, **real**, and **cmplx** functions.

char2num

char2num(*str*)

The char2num function returns a numeric code representing the first byte of *str* or the first character of *str*.

If *str* contains zero bytes, char2num returns NaN.

If *str* contains exactly one byte, char2num returns the value of that byte, treated as a signed byte. For backward compatibility with Igor6, if the input is a single byte in the range 0x80..0xFF, char2num returns a negative number.

If *str* contains more than one byte, char2num returns a number which is the Unicode code point for the first character in *str* treated as UTF-8 text. If *str* does not start with a valid UTF-8 character, char2num returns NaN.

Prior to Igor Pro 7.00, char2num always returned the value of the first byte, treated as a signed byte.

Examples

```
Function DemoChar2Num()
    String str

    str = "A"
    Printf "Single ASCII character: %02X\r", char2num(str)          // Prints 0x41

    str = "ABC"
    Printf "Multiple ASCII characters: %02X\r", char2num(str)       // Prints 0x41

    str = U+2022        // Bullet character
```