

[Accelerate](#) / [vDSP](#) / Absolute and negation functions

API Collection

Absolute and negation functions

Compute the absolute or negated value of each element in a vector.

Topics

Vector absolute functions

```
static func absolute<U>(U) -> [Float]
```

Returns the absolute value of each element in the supplied single-precision vector.

```
static func absolute<U>(U) -> [Double]
```

Returns the absolute value of each element in the supplied double-precision vector.

```
static func absolute<U, V>(U, result: inout V)
```

Calculates the absolute value of each element in the supplied single-precision vector.

```
static func absolute<U, V>(U, result: inout V)
```

Calculates the absolute value of each element in the supplied double-precision vector.

`vDSP_vabsi`

Calculates the absolute value of each element in the supplied integer vector using the specified stride.

`vDSP_vabs`

Calculates the absolute value of each element in the supplied single-precision vector using the specified stride.

`vDSP_vabsD`

Calculates the absolute value of each element in the supplied double-precision vector using the specified stride.

Complex vector absolute functions

```
static func absolute<V>(DSPSplitComplex, result: inout V)
```

Calculates the absolute value of each element in the supplied single-precision complex vector.

```
static func absolute<V>(DSPDoubleSplitComplex, result: inout V)
```

Calculates the absolute value of each element in the supplied double-precision complex vector.

```
vDSP_zvabs
```

Calculates the absolute value of each element in the supplied single-precision complex vector using the specified stride.

```
vDSP_zvabsD
```

Calculates the absolute value of each element in the supplied double-precision complex vector using the specified stride.

Vector negative absolute functions

```
static func negativeAbsolute<U>(U) -> [Float]
```

Returns the negative absolute value of each element in the supplied single-precision vector.

```
static func negativeAbsolute<U>(U) -> [Double]
```

Returns the negative absolute value of each element in the supplied double-precision vector.

```
static func negativeAbsolute<U, V>(U, result: inout V)
```

Calculates the negative absolute value of each element in the supplied single-precision vector.

```
static func negativeAbsolute<U, V>(U, result: inout V)
```

Calculates the negative absolute value of each element in the supplied double-precision vector.

```
vDSP_vnabs
```

Calculates the negative absolute value of each element in the supplied single-precision vector using the specified stride.

vDSP_vnabsD

Calculates the negative absolute value of each element in the supplied double-precision vector using the specified stride.

Vector negation functions

```
static func negative<U>(U) -> [Float]
```

Returns the negative value of each element in the supplied single-precision vector.

```
static func negative<U>(U) -> [Double]
```

Returns the negative value of each element in the supplied double-precision vector.

```
static func negative<U, V>(U, result: inout V)
```

Calculates the negative value of each element in the supplied single-precision vector.

```
static func negative<U, V>(U, result: inout V)
```

Calculates the negative value of each element in the supplied double-precision vector.

vDSP_vneg

Calculates the negative value of each element in the supplied single-precision vector using the specified stride.

vDSP_vnegD

Calculates the negative value of each element in the supplied double-precision vector using specified stride.

Complex vector negation functions

vDSP_zvneg

Calculates the negative value of each element in the supplied complex single-precision vector.

vDSP_zvnegD

Calculates the negative value of each element in the supplied complex double-precision vector.

See Also

Single-vector arithmetic functions

☰ Integration functions

Compute the running sum, Simpson, or trapezoidal integration of a vector.

☰ Clipping, limit, and threshold operations

Apply clipping, limit, or threshold rules to the elements in a vector.

☰ Normalization functions

Compute the mean and standard deviation of a vector and calculate new elements to have a zero mean and a unit standard deviation.

☰ Phase computation functions

Calculate the element-wise phase values, in radians, of a complex vector.

☰ Complex conjugation functions

Calculate the complex conjugate of the elements in a vector.

☰ Vector squaring functions

Compute the square, signed square, or squared magnitude of the elements in a vector.

☰ Fractional part extraction

Truncate the elements of a vector to a fraction.

☰ Zero crossing search

Count and find the zero crossings in a vector.