

# MathArray (Extensions)

**Inherits From:** NSObject

**Declared In:** MathArray/MathArrayExtensions.h, MathArray/MANumber.h

## Category Description

MathArray is a general class for performing mathematical operations on array (1-D vectors, 2-D matrices, etc) of values. See MathArray for a better description of the class. This document describes some convenient extensions to the 'spartan' class definition. Most of these methods are implemented in terms of other methods.

## Method Types

Operating on/with arrays

- maAdd:
- maAnd:
- maDivide:
- maEqual:

- maExponent:
- maGreater:
- maGreaterOrEqual:
- maLess:
- maLessOrEqual:
- maMatrixMultiply:
- maMaximum:
- maMaximumValue
- maMinimum:
- maMinimumValue
- maMod:
- maMultiply:
- maNot
- maNotEqual:
- maOr:
- maSubtract:
- maXor:

Performing functions on arrays

- maAbs
- maArcCos
- maArcSin
- maArcTan
- maCos
- maExp
- maFFt:
- maInvert
- maLogBase10
- maNaturalLog
- maSin

- maSqrt
- maTan

Other functions

- maShift:
- maTotal
- maTranspose
- maWhere

## Instance Methods

**maAbs**

- **maAbs**

Takes the absolute value of the receiver..

**See also:**

**maAdd:**

- **maAdd:***value*

Adds *value* to the receiver.

**See also:**  $\pm$  **maOperate:with:**

**maAnd:**

- **maAnd:***value*

Takes the logical AND of *value* and the receiver.

**See also:**  $\pm$  `maOperate:with:`

**maArcCos**

- `maArcCos`

Takes the inverse or 'arc' cosine of the receiver.

**See also:**  $\pm$  `maPerform:`

**maArcSin**

- `maArcSin`

Takes the inverse or 'arc' sine of the receiver.

**See also:**  $\pm$  `maPerform:`

**maArcTan**

- `maArcTan`

Takes the inverse or 'arc' tangent of the receiver.

**See also:**  $\pm$  `maPerform:`

**maCos**

- `maCos`

Takes the cosine of the receiver.

**See also:**  $\pm$  **maPerform:**

**maDivide:**

- **maDivide:***divisor*

Divides the receiver by *value*.

**See also:**  $\pm$  **maOperate:with:**

**maEqual:**

- **maEqual:***value*

Logically test for the equality of the receiver and *value*.

**See also:**  $\pm$  **maOperate:with:**

**maExp**

- **maExp**

Replaces the receiver by `exp(receiver)`.

**See also:**  $\pm$  **maPerform:**

**maExponent:**

- **maExponent:***exp*

Replaces the receiver by  $(\text{receiver})^{\text{exp}}$ .

**See also:**  $\pm$  `maOperate:with:`

**maFFt:**

- `maFFt:(int)direction`

Performs a Fast-Fourier Transform on the receiver in the specified direction: -1 for a forward transform, +1 for an inverse transform. The receiver is automatically converted to a complex type. NOTE: Currently the FFt can only be performed on arrays that have sizes that are integer powers of two.

**See also:**

**maGreater:**

- `maGreater:value`

Logically tests for the receiver being greater than *value*.

**See also:**  $\pm$  `maOperate:with:`

**maGreaterOrEqual:**

- `maGreaterOrEqual:value`

Logically tests for the receiver being greater than or equal to *value*.

**See also:**  $\pm$  `maOperate:with:`

## **maInvert**

- **maInvert**

Inverts the array.

**See also:**

## **maLess:**

- **maLess:***value*

Logically tests for the receiver being less than *value*.

**See also:**  $\pm$  **maOperate:with:**

## **maLessOrEqual:**

- **maLessOrEqual:***value*

Logically tests for the receiver being less than or equal to *value*.

**See also:**  $\pm$  **maOperate:with:**

## **maLogBase10**

- **maLogBase10**

Take the logarithm (base 10) of the receiver.

**See also:**  $\pm$  **maPerform:**

**maMatrixMultiply:**

- **maMatrixMultiply:(MathArray \*)otherArray**

Performs matrix or vector multiplication between the receiver and *otherArray*. Raises an **MAArrayMismatchException** if the dimension or sizes of the arrays are not compatible..

**See also:**  $\pm$  **maOperate:with:**

**maMaximum:**

- **maMaximum:value**

Replaces each element of the receiver by the maximum of either the receiver or value.

**See also:**  $\pm$  **maOperate:with:**

**maMaximumValue**

- (id <NSNumber,ComplexNumber>)**maMaximumValue**

Returns a new MNumber which contains the maximum value of the array.

**See also:**

**maMinimum:**

- **maMinimum:value**

Replaces each element of the receiver by the minimum of either the receiver or value.

**See also:**  $\pm$  **maOperate:with:**

### **maMinimumValue**

- (id <NSNumber,ComplexNumber>)maMinimumValue

Returns a new MNumber which contains the minimum value of the array.

**See also:**

### **maMod:**

- maMod:*value*

Replaces each element of the receiver with the mathematical MOD of the receiver and *value*.

**See also:** ± maOperate:with:

### **maMultiply:**

- maMultiply:*factor*

Multiplies the receiver by *factor*.

**See also:** ± maOperate:with:

### **maNaturalLog**

- maNaturalLog

Takes the natural logarithm of the receiver.

**See also:** ± maPerform:

**maNot**

- **maNot**

Performs a logical NOT on the receiver.

**See also:**  $\pm$  **maOperate:with:**

**maNotEqual:**

- **maNotEqual:***value*

Logically test for the receiver and *value* being not equal.

**See also:**  $\pm$  **maOperate:with:**

**maOr:**

- **maOr:***value*

Performs a logical OR between the receiver and *value*.

**See also:**  $\pm$  **maOperate:with:**

**maShift:**

- **maShift:**(int \*)*positions*

Shift each value of the array by an amount specified in the array *positions*. If, for instance, the receiver is a two-dimensional array and *positions* = {4,-7}, then each value in the receiver is shifted 4 rows down (forward) and 7 columns back. Values wrap around in the array.

**See also:**

**maSin**

- **maSin**

Takes the sine of the receiver.

**See also:**  $\pm$  **maPerform:**

**maSqrt**

- **maSqrt**

Takes the square root of the receiver.

**See also:**  $\pm$  **maPerform:**

**maSubtract:**

- **maSubtract:***value*

Subtracts *value* from the receiver.

**See also:**  $\pm$  **maOperate:with:**

**maTan**

- **maTan**

Takes the tangent of the receiver.

**See also:**  $\pm$  `maPerform:`

### **maTotal**

- (id <NSNumber,ComplexNumber>) **maTotal**

Returns the sum of every element in the array as an MNumber.

**See also:**

### **maTranspose**

- **maTranspose**

Transposes the array.

**See also:**

### **maWhere**

- **maWhere**

Replaces the receiver by an array which contains the *orderedIndex* values of where the receiver was nonzero. You can use this in more complex expressions to, for instance, return the location of where the receiver is greater than 2:

```
number2 = [NSNumber numberWithInt:2.0];  
c = [[[MA_TEMPORARY(b) maGreater:number2] maWhere]
```

**See also:**  $\pm$  `arrayValues:`

**maXor:**

- **maXor:***value*

Performs the logical XOR of the receiver with *value*.

**See also:**

## **MathArray (ComplexExtensions)**

**Inherits From:** NSObject

**Declared In:** MathArray/MathArrayExtensions.h

### **Catagory Description**

These are extension methods for use with complex numbers.

### **Method Types**

Making complex arrays

+ `maComplexArrayWithReal:imaginary:`

Operating on arrays

- `isComplex`  
- `maConjugate`  
- `maImaginary`  
- `maMagnitude`  
- `maPhase`  
- `maReal`

## Class Methods

**`maComplexArrayWithReal:imaginary:`**

+ `(MathArray *)maComplexArrayWithReal:(MathArray *)realArray  
imaginary:(MathArray *)imagArray`

Returns a new array of complex type made from the combination of the non-complex arrays *realArray* and *imagArray*.

**See also:**

## Instance Methods

**`isComplex`**

- `(BOOL)isComplex`

Returns *YES* if the receiver is complex type.

**See also:**

### **maConjugate**

- **maConjugate**

Replaces the receiver by its complex conjugate. If the array is not complex, it is automatically promoted to a complex type.

**See also:**

### **malmaginary**

- **malmaginary**

Replaces the receiver by its imaginary part. The array is set to 0 if it is not complex.

**See also:**  $\pm$  **maReal**

### **maMagnitude**

- **maMagnitude**

Replaces the receiver by its magnitude, i.e.  $\text{SQRT}(\{Re\}^2 + \{Im\}^2)$ .

**See also:**  $\pm$  **maPhase**

### **maPhase**

- **maPhase**

Replaces the receiver by its phase, i.e.  $\tan^{-1}(Im/Re)$ . The array is set to 0 if it is not complex.

**See also:**  $\pm$  maMagnitude

**maReal**

- maReal

Replaces the receiver by its real part.

**See also:**  $\pm$  malmaginary