Contents

Octave operators

Arithmetic Operators

+	Addition('+') operator can be used to add to numbers or matrices.
-	Subtraction('-') operator can be used to subtract one value / matrix from another value / matrix.
*	Multiplication('*') operator can be used to multiply two numbers or matrices. For matrix multiplication the number of columns in first matrix should be equal to number of rows in second matrix else you would get error like -- 'error: operator *: nonconformant arguments (op1 is 2x4, op2 is 2x4)'
x/y	Right division. Conceptually same as (inverse (y') * x')' But it is calculated without finding inverse of y'.
x\y	Left division. This is conceptually equivalent to the expression inverse (x) * y
.+ +.	Element wise addition

.- -.	Element wise subtraction
.*	Element wise multiplication
./	Element by element right divison
.\	Element by element left division. Each element of y is divided by each corresponding element of x.
x^y x**y	For scalars x and y it returns x to power y
x.^y x.**y	Element by element power operator.
x.'	Transpose
x'	Complex conjugate of transpose. This operator is equivalent to the expression conj (x.')

Comparison Operators

x<y	True if x is less than y
x <= y	True if x is less than equal to y.
x == y	True if x is equal to y
x >= y	True if x is greate than or equal to y
x>y	True if x is greater than y.
x!=y x~=y x <> y	True if x is not equal to y.

Boolean operators

Normal boolean operators

a&b	True if both `a' and `b' are true
a|b	True if either of `a' or `b' is true
!a ~a	True if a is false

Short circuit boolean operators

a&&b	True if both a and b are true. Does not evaluates b if a is false.
a || b	True if either of a or b is true. Does not evaluates b if a is true.

Assignment operators

=	Normal assignment operators
+=	Shorthand addition and assignment operator
-=	Shorthand subtraction and assignment operator
*=	Shorthand multiplication and assignment operator
/=	Shorthand division and assignment operator

Few important points regarding assignment operators:

- Assignment of a scalar to an indexed matrix sets all of the elements that are referenced by the indices to the scalar value. For example, if a is a matrix with at least two columns, then
  a(:, 2) = 5
  sets all the elements in the second column of a to 5.

- Assigning an empty matrix `[]' works in most cases to allow you to delete rows or columns of matrices and vectors. For example

A (3, :) = []
deletes the third row of A

- An assignment is an expression, so it has a value. Thus, z = 1 as an expression has the value 1. One consequence of this is that you can write multiple assignments together:
  x = y = z = 0
  stores the value 0 in all three variables.
- In expressions like this, the number of values in each part of the expression need not match. For example, the expression
  [a, b] = [u, s, v] = svd (a)
  is equivalent to
  [u, s, v] = svd (a)
  a = u
  b = s
  The number of values on the left side of the expression can, however, not exceed the number of values on the right side.

Increment operators

++x	This expression increments the variable x. The value of the expression is the new value of x. It is equivalent to the expression x = x + 1.
--x	This expression decrements the variable x. The value of the expression is the new value of x. It is equivalent to the expression x = x - 1.
x++	This expression causes the variable x to be incremented. The value of the expression is the old value of x.
x--	This expression causes the variable x to be decremented. The value of the expression is the old value of x.

Operator Precedence

Here is a table of the operators in Octave, in order of increasing precedence.

```
statement separators
    `;', `,'.
assignment
    `=', `+=', `-=', `*=',`/='. This operator groups right to left.
logical "or" and "and"
    `||', `&&'.
element-wise "or" and "and"
```

```
    `|', `&'.
relational
    `<', `<=', `==', `>=', `>', `!=', `~=', `<>'.
colon
    `:'.
add, subtract
    `+', `-'.
multiply, divide
    `*', `/', `\', `.\', `.*', `./'.
transpose
    `'', `.''
unary plus, minus, increment, decrement, and ``not''
    `+', `-', `++', `--', `!', `~'.
exponentiation
    `^', `**', `.^', `.**'.
```