range( ) — Vector over specified range

Description Syntax Remarks and examples Conformability Diagnostics Also see

Description

`range(`*a*`,` *b*`,` *delta*`)` returns a column vector going from *a* to *b* in steps of `abs(`*delta*`)` ($b \geq a$) or $-$`abs(`*delta*`)` ($b < a$).

`rangen(`*a*`,` *b*`,` *n*`)` returns a `round(`*n*`)` $\times$ 1 column vector going from *a* to *b* in `round(`*n*`)-1` steps. *a* may be less than, equal to, or greater than *b*.

Syntax

numeric colvector `range(`*a*`,` *b*`,` *numeric scalar delta*`)`

numeric colvector `rangen(`*a*`,` *b*`,` *real scalar n*`)`

where *a* and *b* are numeric scalars.

Remarks and examples

`range(0, 1, .25)` returns `(0 \ .25 \ .5 \ .75 \ 1)`. The sign of the third argument does not matter; `range(0, 1, -.25)` returns the same thing. `range(1, 0, .25)` and `range(1, 0, -.25)` return `(1 \ .75 \ .5 \ .25 \ 0)`.

`rangen(0, .5, 6)` returns `(0 \ .1 \ .2 \ .3 \ .4 \ .5)`. `rangen(.5, 0, 6)` returns `(.5 \ .4 \ .3 \ .2 \ .1 \ 0)`.

`range()` and `rangen()` may be used with complex arguments. `range(1, 1i, .4)` returns `(1 \ .75+.25i \ .5+.5i \ .25+.75i \ 1i)`. `rangen(1, 1i, 5)` returns the same thing. For `range()`, only the distance of *delta* from zero matters, so `range(1, 1i, .4i)` would produce the same result, as would `range(1, 1i, .25+.312i)`.

Conformability

`range(`*a*`,` *b*`,` *delta*`)`:

a:	1×1
b:	1×1
delta:	1×1
result:	1×1, if $a = b$
	`max(1+abs(`*b*$-$*a*`)/abs(`*delta*`),2)` $\times$ 1, otherwise

`rangen(`*a*`,` *b*`,` *n*`)`:

a:	1×1
b:	1×1
n:	$n \times 1$
result:	`round(`*n*`)` $\times 1$

Diagnostics

`range(`*a*`,` *b*`,` *delta*`)` aborts with error if *a*, *b*, or *delta* contains missing, if `abs(`*b*`−`*a*`)/abs(`*delta*`)` results in overflow, or if `1+abs(`*b*`−`*a*`)/abs(`*delta*`)` results in a vector that is too big given the amount of memory available.

`range(`*a*`,` *b*`,` *delta*`)` returns a 1×1 result when $a = b$. In all other cases, the result is 2×1 or longer.

`rangen(`*a*`,` *b*`,` *n*`)` aborts with error if `round(`*n*`)` is less than 0 or missing.

Also see

[M-4] **Standard** — Functions to create standard matrices