

[Scipy.org \(https://scipy.org/\)](https://scipy.org/) [Docs \(https://docs.scipy.org/\)](https://docs.scipy.org/)

[SciPy v1.1.0 Reference Guide \(../index.html\)](#)

[Integration and ODEs \(**scipy.integrate**\) \(../integrate.html\)](#)

[index \(../genindex.html\)](#) [modules \(../py-modindex.html\)](#) [next \(scipy.integrate.romb.html\)](#)

[previous \(scipy.integrate.cumtrapz.html\)](#)

scipy.integrate.simps

scipy.integrate.simps(*y*, *x=None*, *dx=1*, *axis=-1*, *even='avg'*) [[source](#)]

(<https://github.com/scipy/scipy/blob/v1.1.0/scipy/integrate/quadrature.py#L334-L456>)

Integrate $y(x)$ using samples along the given axis and the composite Simpson's rule. If x is `None`, spacing of dx is assumed.

If there are an even number of samples, N , then there are an odd number of intervals ($N-1$), but Simpson's rule requires an even number of intervals. The parameter `'even'` controls how this is handled.

Parameters: *y* : *array_like*

Array to be integrated.

x : *array_like, optional*

If given, the points at which y is sampled.

dx : *int, optional*

Spacing of integration points along axis of y . Only used when x is `None`. Default is 1.

axis : *int, optional*

Axis along which to integrate. Default is the last axis.

even : *str {'avg', 'first', 'last'}, optional*

'avg' : Average two results: 1) use the first $N-2$ intervals with

a trapezoidal rule on the last interval and 2) use the last $N-2$ intervals with a trapezoidal rule on the first interval.

'first' : Use Simpson's rule for the first $N-2$ intervals with

a trapezoidal rule on the last interval.

'last' : Use Simpson's rule for the last $N-2$ intervals with a

trapezoidal rule on the first interval.

See also:

quad ([scipy.integrate.quad.html#scipy.integrate.quad](#)) adaptive quadrature using QUADPACK

romberg ([scipy.integrate.romberg.html#scipy.integrate.romberg](#)) adaptive Romberg quadrature

quadrature ([scipy.integrate.quadrature.html#scipy.integrate.quadrature](#)) adaptive Gaussian quadrature

fixed_quad ([scipy.integrate.fixed_quad.html#scipy.integrate.fixed_quad](#)) fixed-order Gaussian quadrature

dblquad ([scipy.integrate.dblquad.html#scipy.integrate.dblquad](#)) double integrals