https://scipy.org/about.html

Scientific Computing Tools for Python

SciPy refers to several related but distinct entities:

- The *SciPy Stack*, a collection of open source software for scientific computing in Python, and particularly a specified
- The *community* of people who use and develop this stack.
- Several *conferences* dedicated to scientific computing in Python - SciPy, EuroSciPy and SciPy.in.
- The SciPy library, one component of the SciPy stack, providing many numerical routines.

The SciPy Stack

Core Packages

- Python, a general purpose programming language. It is interpreted and dynamically typed and is very suited for int powerful enough to write large applications in.
- NumPy, the fundamental package for numerical computation. It defines the numerical array and matrix types and
- The SciPy library, a collection of numerical algorithms and domain-specific toolboxes, including signal processing, o
- Matplotlib, a mature and popular plotting package, that provides publication-quality 2D plotting as well as rudimen
- pandas, providing high-performance, easy to use data structures.
- SymPy, for symbolic mathematics and computer algebra.
- IPython, a rich interactive interface, letting you quickly process data and test ideas. The **IPython notebook** works in computation in an easily reproducible form.
- nose, a framework for testing Python code.

https://docs.scipy.org/doc/scipy/reference/

Tutorial

Tutorials with worked examples and background information for most SciPy submodules.

- SciPy Tutorial
  - Introduction
  - Basic functions
  - Special functions (`scipy.special`)
  - Integration (`scipy.integrate`)
  - Optimization (`scipy.optimize`)
  - Interpolation (`scipy.interpolate`)
  - Fourier Transforms (`scipy.fftpack`)
  - Signal Processing (`scipy.signal`)
  - Linear Algebra (`scipy.linalg`)
  - Sparse Eigenvalue Problems with ARPACK
  - Compressed Sparse Graph Routines (`scipy.sparse.csgraph`)
  - Spatial data structures and algorithms (`scipy.spatial`)
  - Statistics (`scipy.stats`)
  - Multidimensional image processing (`scipy.ndimage`)
  - File IO (`scipy.io`)

https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html

Table Of Contents

https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html#scipy.integrate.qua

scipy.integrate.quad

scipy.integrate.**quad**(*func, a, b, args=(), full_output=0, epsabs=1.49e-08, epsrel=1.49e-08, limit=50, wvar=None, wopts=None, maxp1=50, limlst=50*)

Compute a definite integral.

```
from scipy.integrate import quad
import numpy as np

def myfunc1(x):
    return 3*np.exp(-2*x)

def myfunc2(x,a,b):
    return a*np.exp(-b*x)

def main():
    ival,error = quad(myfunc1,0,1)
    print(ival,error)

    ival,error = quad(myfunc2,0,1,args=(3,2))
    print(ival,error)

    j=3; k=2
    ival,error = quad(myfunc2,0,1,args=(j,k))
    print(ival,error)

main()
```

$$\int_0^1 3e^{-2x}\,dx = 1.297$$

```
C:\Users\Delahoussaye\Anaconda3\python.exe
1.296997075145081 1.4399560156975533e-14
1.296997075145081 1.4399560156975533e-14
1.296997075145081 1.4399560156975533e-14

Process finished with exit code 0
```