

RWEPA | 程式設計 Python

SciPy 模組 - 數值積分, 內插法

Business Data Analytics

- 按訂閱、讚、開啟小鈴鐺
- <https://youtube.com/@alan9956>
- <http://rwepa.blogspot.com/>



大綱

- 數值積分
- 內插法
- Q & A

數值積分

(Numerical integration)

scipy.integrate 三大功能

- General integration (quad) 積分- quad
- 數值積分
 - trapezoidal rule 梯形法則 - **trapz**
 - Simpson's rule 辛浦森法則 - **simps**
- 微分方程 ordinary differential equations - odeint

梯形法則 Trapezoidal rule

- 若 f 在區間 $[a, b]$ 連續，則

$$\int_a^b f(x)dx = \frac{\Delta x}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)]$$

其中

$$\Delta x = \frac{b-a}{n}, a = x_0, b = x_n, n = \text{梯形數目}$$

$$x_1 = x_0 + \Delta x, x_2 = x_1 + \Delta x, x_3 = x_2 + \Delta x, \dots$$

梯形法則 - 範例

Use the Trapezoidal Rule to approximate

$$\int_0^{\pi} \sin x \, dx.$$

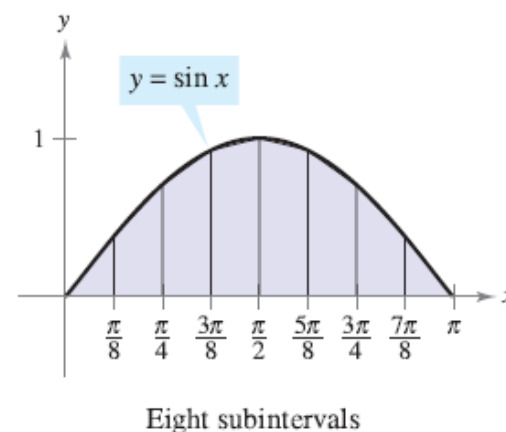
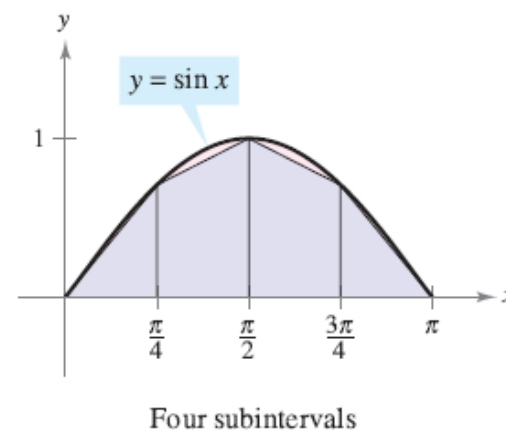
Compare the results for $n = 4$ and $n = 8$, as shown in Figure 4.44.

Solution When $n = 4$, $\Delta x = \pi/4$, and you obtain

$$\begin{aligned} \int_0^{\pi} \sin x \, dx &\approx \frac{\pi}{8} \left(\sin 0 + 2 \sin \frac{\pi}{4} + 2 \sin \frac{\pi}{2} + 2 \sin \frac{3\pi}{4} + \sin \pi \right) \\ &= \frac{\pi}{8} (0 + \sqrt{2} + 2 + \sqrt{2} + 0) = \frac{\pi(1 + \sqrt{2})}{4} \approx 1.896. \end{aligned}$$

When $n = 8$, $\Delta x = \pi/8$, and you obtain

$$\begin{aligned} \int_0^{\pi} \sin x \, dx &\approx \frac{\pi}{16} \left(\sin 0 + 2 \sin \frac{\pi}{8} + 2 \sin \frac{\pi}{4} + 2 \sin \frac{3\pi}{8} + 2 \sin \frac{\pi}{2} \right. \\ &\quad \left. + 2 \sin \frac{5\pi}{8} + 2 \sin \frac{3\pi}{4} + 2 \sin \frac{7\pi}{8} + \sin \pi \right) \\ &= \frac{\pi}{16} \left(2 + 2\sqrt{2} + 4 \sin \frac{\pi}{8} + 4 \sin \frac{3\pi}{8} \right) \approx 1.974. \end{aligned}$$



辛浦森法則

- 若 f 在區間 連續，則

$$\int_a^b f(x)dx = \frac{\Delta x}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

其中

$$\Delta x = \frac{b-a}{n}, a = x_0, b = x_n, n \text{ 必須為偶數}$$

$$x_1 = x_0 + \Delta x, x_2 = x_1 + \Delta x, x_3 = x_2 + \Delta x, \dots$$

辛浦森法則 - 範例

Use Simpson's Rule to approximate

$$\int_0^{\pi} \sin x \, dx.$$

Compare the results for $n = 4$ and $n = 8$.

Solution When $n = 4$, you have

$$\int_0^{\pi} \sin x \, dx \approx \frac{\pi}{12} \left(\sin 0 + 4 \sin \frac{\pi}{4} + 2 \sin \frac{\pi}{2} + 4 \sin \frac{3\pi}{4} + \sin \pi \right) \approx 2.005.$$

When $n = 8$, you have $\int_0^{\pi} \sin x \, dx \approx 2.0003$.

integrate.trapz 梯形法則

integrate.simps 辛浦森法則

```
In [1]: from scipy.integrate import trapz, simps
```

```
In [2]: import numpy as np
```

```
In [3]: def f1(x):  
...:     return np.sin(x)
```

```
In [4]: x1 = np.linspace(0, np.pi, 5)  
...: x1
```

```
Out[4]: array([0.          , 0.78539816, 1.57079633, 2.35619449, 3.14159265])
```

```
In [5]: trapint = trapz(f1(x1), x1)  
...: print(trapint)  
1.8961188979370398
```

1 2

```
In [6]: sinint = simps(f1(x1), x1)  
...: print(sinint)  
2.0045597549844207
```

trapz, simps

```
In [7]: x2 = np.linspace(0, np.pi, 9)
...: x2
```

```
Out[7]:
array([0.          , 0.39269908, 0.78539816, 1.17809725, 1.57079633,
       1.96349541, 2.35619449, 2.74889357, 3.14159265])
```

```
In [8]: trapint = trapz(f1(x2), x2)
...: print(trapint)
1.9742316019455508
```

```
In [9]: sinint = simps(f1(x2), x2)
...: print(sinint)
2.0002691699483877
```

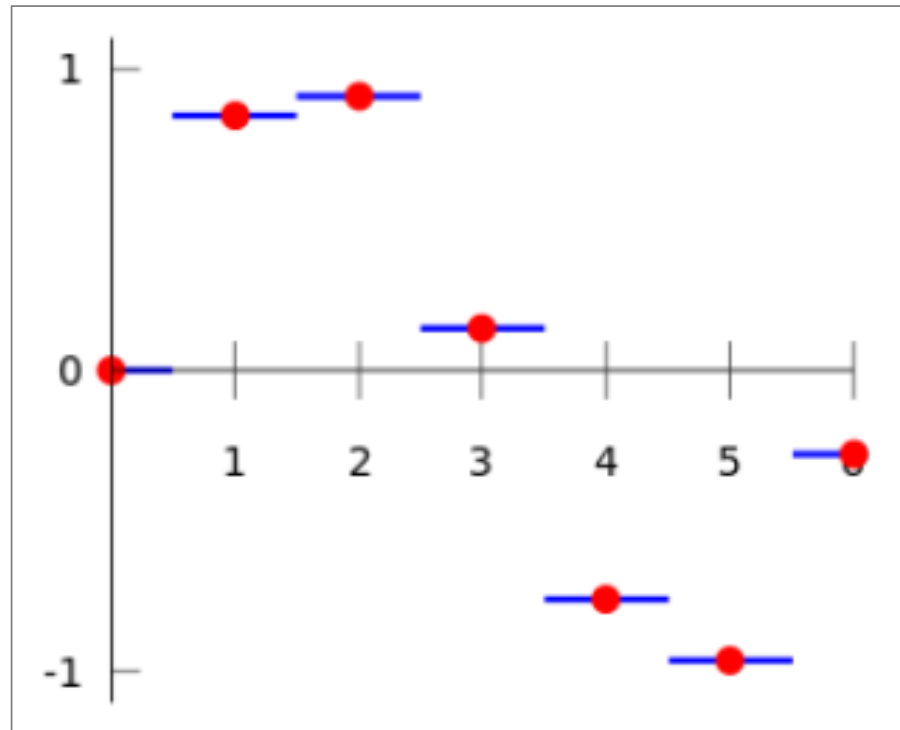
內插法

(Interpolation)

內插法

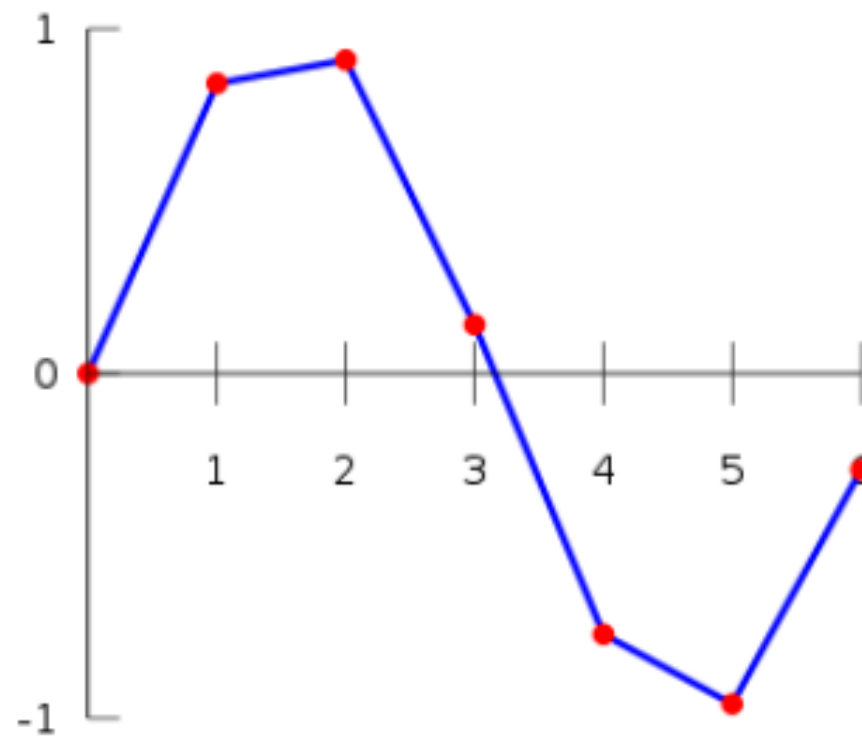
- 在數值分析的數學領域，內插法是一種估計，在一組已知數據點範圍內構造新數據點的方法。
- 求解科學和工程的問題時，通常有許多數據點藉由採樣、實驗等方法獲得，這些數據可能代表了有限個數值函數。而根據這些數據，我們希望得到一個連續的函數（也就是曲線）；或者更密集的離散方程式與已知數據互相吻合，這個過程叫做擬合（fitting）。
- 常用內插法技術
 - Nearest-neighbor interpolation 近鄰插值法
 - Linear interpolation 線性插值法
 - Polynomial interpolation 多項式插值法
 - Spline interpolation 樣條插值法

1.近鄰插值法

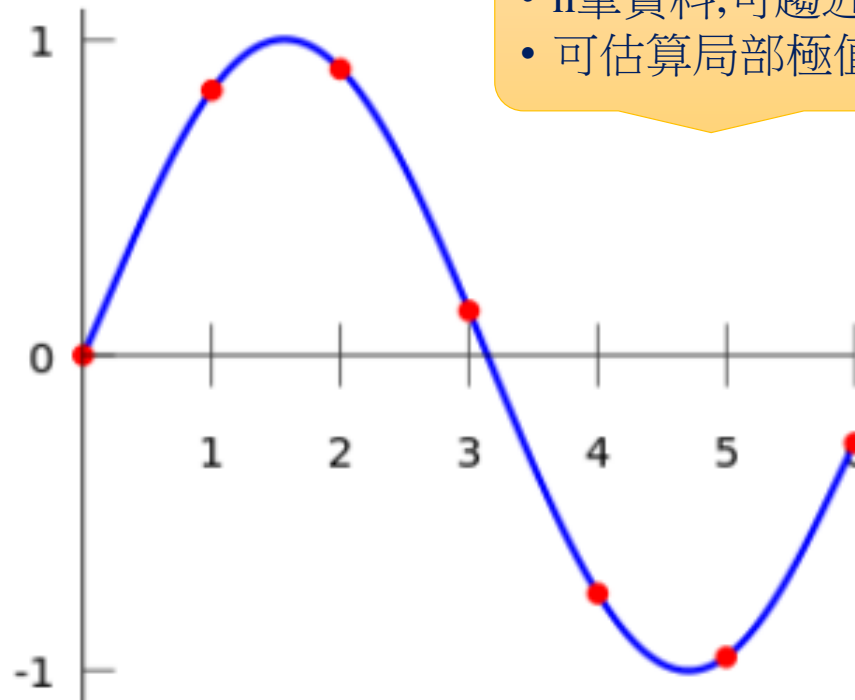


紅色:原始資料

2. 線性插值法



3. 多項式插值法

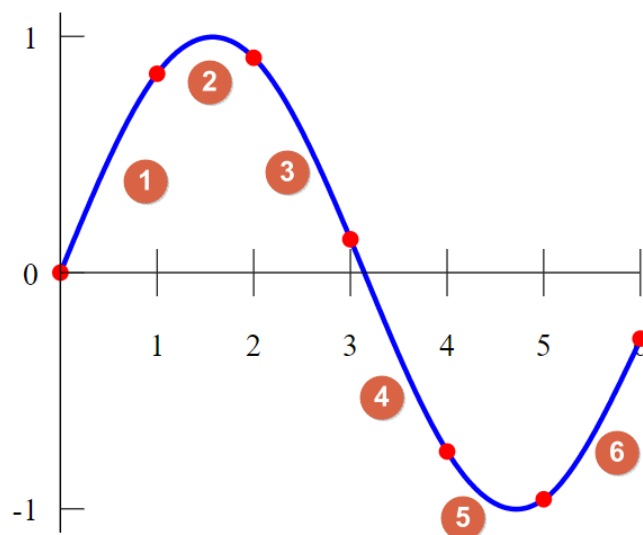


- n筆資料,可趨近於(n-1)階多項式
- 可估算局部極值

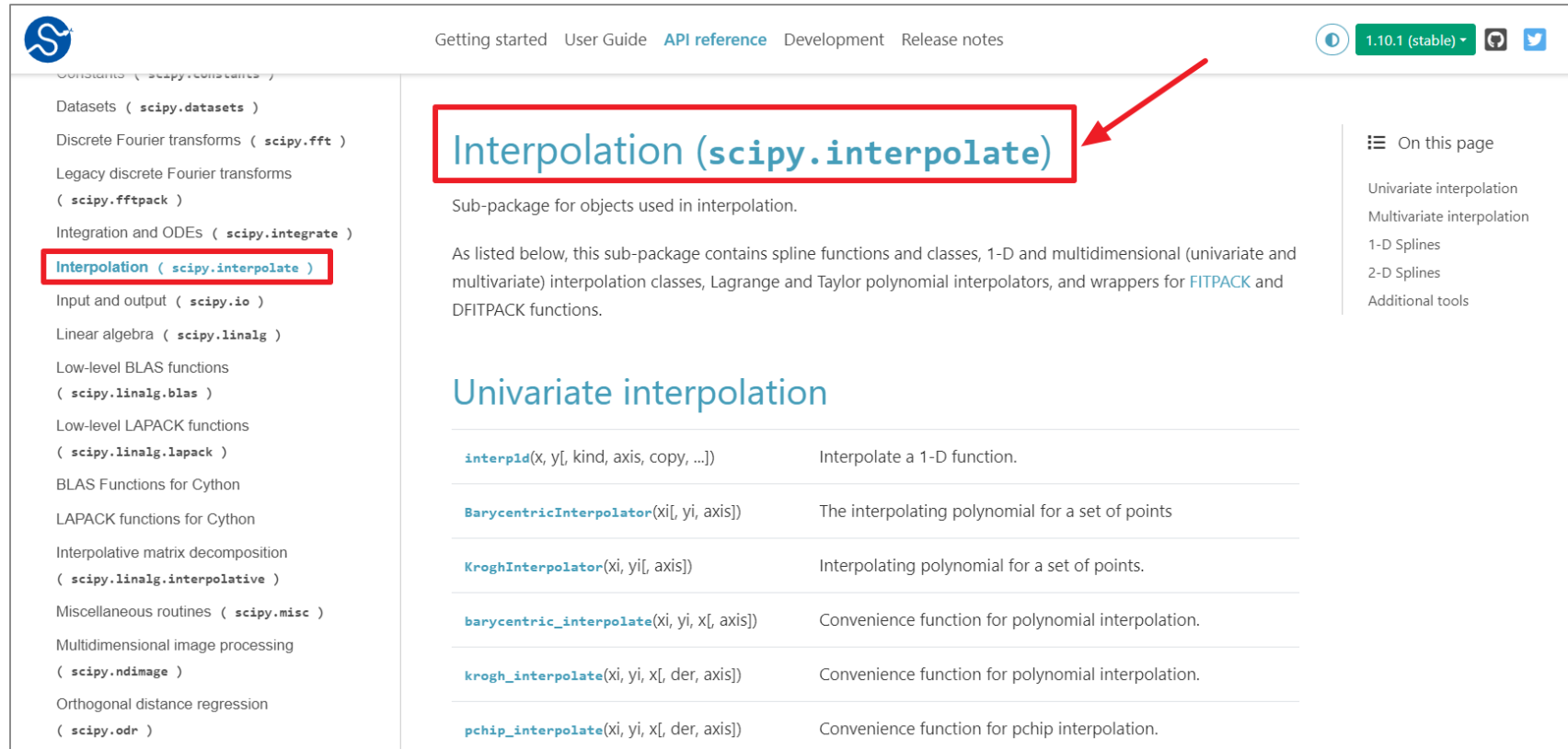
$$f(x) = -0.0001521x^6 - 0.003130x^5 + 0.07321x^4 - 0.3577x^3 + 0.2255x^2 + 0.9038x.$$

4. 樣條插值法

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x, & \text{if } x \in [0, 1], \\ -0.01258x^3 - 0.4189x^2 + 1.4126x - 0.1396, & \text{if } x \in [1, 2], \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623, & \text{if } x \in [2, 3], \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381, & \text{if } x \in [3, 4], \\ 0.05375x^3 - 0.2450x^2 - 1.2756x + 4.8259, & \text{if } x \in [4, 5], \\ -0.1871x^3 + 3.3673x^2 - 19.3370x + 34.9282, & \text{if } x \in [5, 6]. \end{cases}$$



Interpolation (scipy.interpolate)



Getting started User Guide **API reference** Development Release notes

1.10.1 (stable)

Interpolation (scipy.interpolate)

Sub-package for objects used in interpolation.

As listed below, this sub-package contains spline functions and classes, 1-D and multidimensional (univariate and multivariate) interpolation classes, Lagrange and Taylor polynomial interpolators, and wrappers for FITPACK and DFITPACK functions.

Univariate interpolation

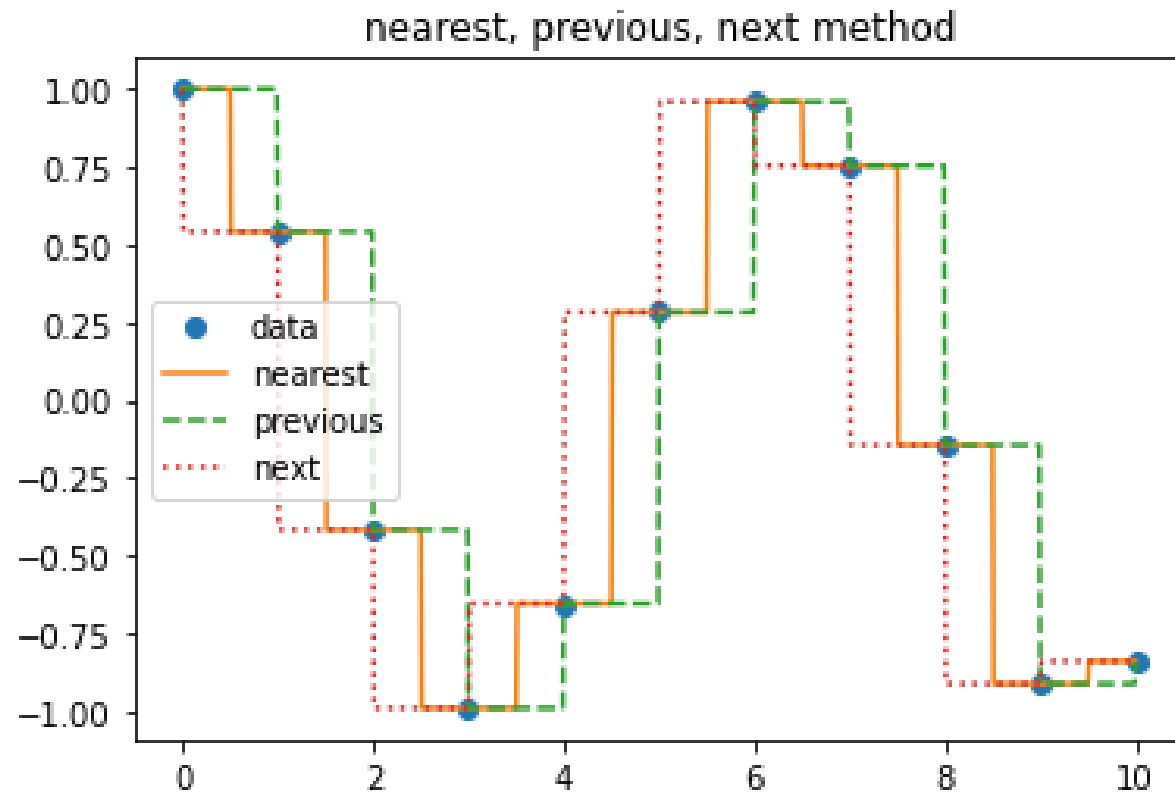
<code>interp1d(x, y[, kind, axis, copy, ...])</code>	Interpolate a 1-D function.
<code>BarycentricInterpolator(xi[, yi, axis])</code>	The interpolating polynomial for a set of points
<code>KroghInterpolator(xi, yi[, axis])</code>	Interpolating polynomial for a set of points.
<code>barycentric_interpolate(xi, yi, x[, axis])</code>	Convenience function for polynomial interpolation.
<code>krogh_interpolate(xi, yi, x[, der, axis])</code>	Convenience function for polynomial interpolation.
<code>pchip_interpolate(xi, yi, x[, der, axis])</code>	Convenience function for pchip interpolation.

On this page

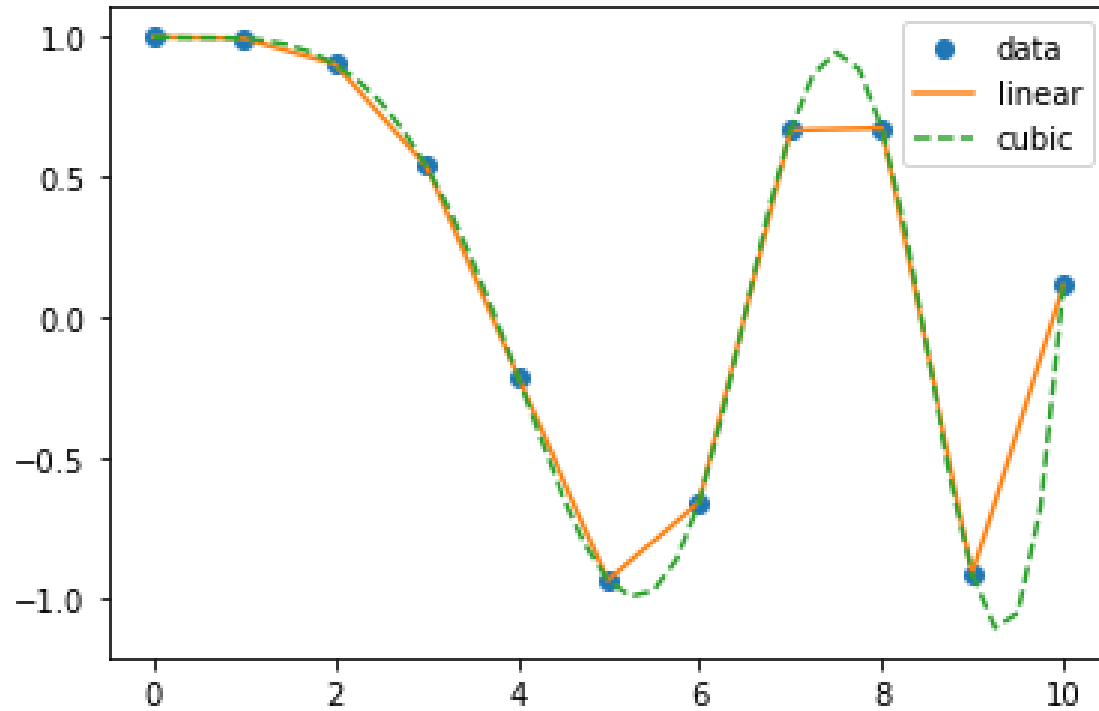
- Univariate interpolation
- Multivariate interpolation
- 1-D Splines
- 2-D Splines
- Additional tools

<https://docs.scipy.org/doc/scipy/reference/interpolate.html>

近鄰插值法



interp1d – cubic (續)



謝謝您的聆聽

Q & A



李明昌

alan9956@gmail.com

<http://rwepa.blogspot.tw/>