顯示程式碼

```
IPython==7.34.0
sympy== 1.13.1
```

簡介

泰勒級數用無限項連加式 -- 級數來表示一個函數，這些相加的項由函數在某一點的導數求得。（WIKI）主要用來將複雜的函數近似為多項式的形式。

泰勒級數在金融領域的應用主要集中在金融產品的定價、風險管理、和數值計算方面，特別是在衍生品定價和投資組合風險評估中。

1. 衍生品訂價
   在期權定價中，泰勒級數可以用於逼近複雜的非線性函數，例如 Black-Scholes 模型。由於許多金融模型的解析解難以求得，使用泰勒級數能夠將複雜的非線性關係分解為較簡單的線性和非線性項，從而更易於分析。例如，假設期權價格對標的資產價格波動敏感，通過二階或三階泰勒級數可以逼近期權價格的變化，幫助理解敏感性和風險（如 delta、gamma 等）。
2. 風險管理
   在投資組合風險管理中，泰勒級數用於計算組合資產價格變動的敏感度（Greeks）。這種方法通過對小的價格變動使用泰勒級數來近似估計不同風險指標的變動，例如價格對利率、波動率的變動，以及資產價格變動對組合的影響。特別是在對非線性收益的投資組合中，高階項可以用於更準確地估計風險。
3. 蒙地卡羅
   泰勒級數也常用於蒙地卡羅中的加速收斂。許多金融模型，特別是隨機漫步模型，往往需要大量模擬才能得出精確結果。通過使用泰勒級數來逼近模型中的複雜函數，可以減少需要模擬的次數，從而提升計算效率。
4. 固定收益產品的凸性調整
   在債券投資中，投資者不僅關注債券的存續期間對利率變動的敏感性，還關注債券的凸性。泰勒展開可以幫助描述債券價格對利率變動的二階近似，進一步調整存續期間的影響，稱為「凸性調整」，用於更精確地評估利率變動對債券價格的影響。
5. 金融模型校正
   一些金融模型（如 GARCH 模型）的預測可能會偏離實際市場行為。在這些情況下，泰勒級數可以用於**校正模型的預測結果**。例如，可以通過泰勒級數對模型中的誤差進行修正，從而提高預測的準確性。

前置知識

- 極限
  函數在自變數無限變大或無限變小或在某個區間時所接近的值。

- 微分導數
  導數反映了函數在某點的變化率，比如直線的斜率。對於函數 $f(x)$，它在點 a 的一階導數 $f'(a)$ 表示 $f(x)$ 在點 a 的變化速率。
- 高階導數
  高階導數是指對函數進行多次微分得到的結果。比如，$f''(x)$ 是 $f(x)$ 的二階導數，表示函數在該點的加速度。三階導數、四階導數等可以依此類推。
- 多項式展開
- 微分均值定理

公式

假設有一個函數 $f(x)$，希望在點 a 使用多項式表達它的變化

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2} + \frac{f'''(a)(x-a)^3}{2 \times 3} + \cdots \Longrightarrow \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^2$$

- n 為導數階數

證明

函數 $f(x)$ 在點 a 的泰勒級數

多項式展開

$$f(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + \cdots c_n(x-a)^n + R_n(x)$$

- $R_n(x)$ 為展開 n 次的誤差項

計算係數

1. c_0
   令 $x = a$ 則
   $$f(a) = c_0$$
2. c_1
   對 $f(x)$ 進行一次微分得到導函數
   $$f'(x) = 1 \times c1 + (1 \times 2)c_2(x-a) + (1 \times 3)c_3(x-a)^2 + \cdots$$

令 $x=a$

$$f'(a)=c_1,\ c1=\frac{f'(x)}{1}=\frac{f'(x)}{1!}$$

3. c_2 對 $f'(x)$ 進行一次微分得到導函數

$$f''(x)=(1\times 2)c_2+(1\times 3)c_3(x-a)^2+\cdots$$

令 $x=a$

$$f''(a)=(1\times 2)c_2,\ c2=\frac{f''(a)}{1\times 2}=\frac{f''(a)}{2!}$$

4. c_3
對 $f''(a)$ 進行一次微分得到導函數

$$f'''(a)=(1\times 2\times 3)c_3+\cdots$$

令 $x=a$

$$f'''(a)=(1\times 2\times 3)c_3,\ c2=\frac{f'''(a)}{1\times 2\times 3}=\frac{f'''(a)}{3!}$$

5. 推論 c_n

$$c_n=\frac{f^{(n)}(a)}{n!}$$

6. 代回 $f(x)$

$$f(x)=f(a)+\frac{f'(a)}{1!}(x-a)^1+\frac{f''(a)}{2!}(x-a)^2+\frac{f'''(a)}{3!}(x-a)^3+\cdots+\frac{f^{(n)}(a)}{n!}(x-a)^n+R_n(x)$$

結論

在有限次下

$$f(x)=\sum_0^n\frac{f^{(n)}(a)}{n!}(x-a)^n+R_n(x)$$

餘項使用均值定理轉換為

$$R_n(x)=\frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{(n+1)},\ \text{where } z\in(x,a)$$

因此當 $n \to \infty$, $R_n(x) \to 0$ 時，級數收斂於 $f(x)$

$$f(x) = \sum_{0}^{n} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

程式版

計算 $f(x) = e^x$ 第 n 個泰勒數列

```
x, a, z = sympy.symbols('x a z')
f = sympy.exp(x)
display(Latex('f(x) = ' + sympy.printing.latex(f)))
```

$f(x) = e^x$

$$f(x) = \sum_{0}^{n} \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n(x)$$

```
# 數列
n_period = 5
taylor_serie = 0

for n in range(n_period):
    f_diff = f.diff(x, n).subs(x, a)
    term = (f_diff / sympy.factorial(n)) * (x - a)**n

    n_str = str(n)
    display(Latex(r'\frac{f^{(' + n_str + ')}(a)}{' + n_str + '!}(x-a)^' + n_str + '=' + sympy.printing.latex(term)))
    print()

    taylor_serie += term

print('-' * 60)
display(taylor_serie)
```

$\frac{f^{(0)}(a)}{0!}(x-a)^0 = e^a$

$\frac{f^{(1)}(a)}{1!}(x-a)^1 = (-a+x)\,e^a$

$\frac{f^{(2)}(a)}{2!}(x-a)^2 = \frac{(-a+x)^2 e^a}{2}$

$\frac{f^{(3)}(a)}{3!}(x-a)^3 = \frac{(-a+x)^3 e^a}{6}$

$\frac{f^{(4)}(a)}{4!}(x-a)^4 = \frac{(-a+x)^4 e^a}{24}$

$$\frac{(-a+x)^4 e^a}{24} + \frac{(-a+x)^3 e^a}{6} + \frac{(-a+x)^2 e^a}{2} + (-a+x)\,e^a + e^a$$

$$R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{(n+1)},$$
$$\text{where } z \in (x, a)$$

```
# 餘項
remainder_term = (f.diff(x, n_period).subs(x, z) / sympy.factorial(n_period)) * (x - a)**(n_period)
display(remainder_term)
```

$$\frac{(-a+x)^5 e^z}{120}$$

```
# 解答
display(sympy.series(f, x, a, n=n_period))
```

$$e^a + (-a+x)e^a + \frac{(-a+x)^2 e^a}{2} + \frac{(-a+x)^3 e^a}{6} + \frac{(-a+x)^4 e^a}{24} + O\left((-a+x)^5; x \to a\right)$$

應用

歐式選擇權

$$C = S \cdot N(d_1) + L \times e^{-rT} N(d_2)$$

- C： 選擇權有效期合理價格
- L： 選擇權有效期履約價格
- S： 標的現價
- r： 無風險利率
- T： 選擇權有效期
- $N(d)$： d 事件常態分佈的累積分佈函數值。
- $d1$：
$$\frac{ln\frac{S}{L} + rT + 0.5\sigma^2 T}{\sigma\sqrt{T}}$$
- $d2$：
$$d1 - \sigma\sqrt{T} = \frac{ln\frac{S}{L} + rT - 0.5\sigma^2 T}{\sigma\sqrt{T}}$$
- σ^2： 年化方差

使用泰勒級數表示在 $S = S_0$ 時的狀況

```
S, S0, L, r, T, sigma = sympy.symbols('S S0 L r T sigma')

d1 = (sympy.ln(S / L) + (r + 0.5 * sigma**2) * T) / (sigma * sympy.sqrt(T))
d2 = d1 - sigma * sympy.sqrt(T)

# 定義累積分佈函數 N(d1) 和 N(d2) 的近似
N_d1_approx = (sympy.erf(d1 / sympy.sqrt(2)) / 2 + 0.5)
N_d2_approx = (sympy.erf(d2 / sympy.sqrt(2)) / 2 + 0.5)

# 定義 Black-Scholes 看漲期權價格公式，使用近似的 N(d1) 和 N(d2)
C = S * N_d1_approx - L * sympy.exp(-r * T) * N_d2_approx
print('Buy Call:')
display(C)

print('-' * 60)
# 對選擇權價格公式進行泰勒展開，展開點 S = S0
taylor = sympy.series(C, S, S0, n=3)
taylor
```

```
Buy Call:
```

$$-L\left(\frac{\operatorname{erf}\left(\frac{\sqrt{2}\left(-\sqrt{T}\sigma+\frac{T\left(r+0.5\sigma^2\right)+\log\left(\frac{S}{L}\right)}{\sqrt{T}\sigma}\right)}{2}\right)}{2}+0.5\right)e^{-Tr}+S\left(\frac{\operatorname{erf}\left(\frac{\sqrt{2}\left(T\left(r+0.5\sigma^2\right)+\log\left(\frac{S}{L}\right)\right)}{2\sqrt{T}\sigma}\right)}{2}+0.5\right)$$

```
------------------------------------------------------------
```

$$(S-S_0)\left(-\frac{0.5\sqrt{2}L\left(\frac{S_0}{L}\right)^{0.5}e^{-0.5Tr}e^{-0.125T\sigma^2}e^{-\frac{0.5\log\left(\frac{S_0}{L}\right)^2}{T\sigma^2}}e^{-\frac{0.5Tr^2}{\sigma^2}}e^{-\frac{1.0r\log\left(\frac{S_0}{L}\right)}{\sigma^2}}}{\sqrt{\pi}S_0\sqrt{T}\sigma}+\frac{\operatorname{erf}\left(\frac{0.5\left(1.0\sqrt{2}T\cdots\right.}{\cdots}\right.}{\cdots}\right.$$

$$(S-S_0)^2\left(\frac{0.25\sqrt{2}Lr\left(\frac{S_0}{L}\right)^{0.5}e^{-0.5Tr}e^{-0.125T\sigma^2}e^{-\frac{0.5\log\left(\frac{S_0}{L}\right)^2}{T\sigma^2}}e^{-\frac{0.5Tr^2}{\sigma^2}}e^{-\frac{1.0r\log\left(\frac{S_0}{L}\right)}{\sigma^2}}}{\sqrt{\pi}S_0^2\sqrt{T}\sigma^3}+\frac{0.125\sqrt{2}L\left(\cdots\right.}{\cdots}\right.$$

$$0.5S_0+\frac{S_0\operatorname{erf}\left(\frac{0.5\left(1.0\sqrt{2}Tr+0.5\sqrt{2}T\sigma^2+1.0\sqrt{2}\log\left(\frac{S_0}{L}\right)\right)}{\sqrt{T}\sigma}\right)}{2}-0.5Le^{-Tr}-$$

$$\frac{Le^{-Tr}\operatorname{erf}\left(\frac{0.5\left(1.0\sqrt{2}Tr-0.5\sqrt{2}T\sigma^2+1.0\sqrt{2}\log\left(\frac{S_0}{L}\right)\right)}{\sqrt{T}\sigma}\right)}{2}+O\left((S-S_0)^3;S\rightarrow S_0\right)$$

其中包含價格對資產變化的一階導數 Δ 與二階導數 Γ

```
# 設定測試值進行替代

subs = {
    L: 100,
    r: 0.05,
    T: 1,
    sigma: 0.2,
}

taylor_subs = taylor.subs(S0, 100).subs(subs)
taylor_subs.evalf().simplify()
```

$$40.5776051838582 - 1.23937108340907S + 0.00938100867292345S^2 + O\left((S - 100)^3; S \to 100\right)$$