科学計算研究室 Pythonゼミ　フォローアップ

～5. Lagrange補間～

2021-03-17 福田　浩

1　Python スクリプト

基本的には C++のソースファイルを 1 行毎 Python に変換している．Python の変換する上での注意点は以下．

- Python で配列の初期化には `a = [0 for i in range(10)]` を使用
- `xx, yy = map(float, input().split(" "))` により，x 座標, y 座標の組を取得し，`xx, yy` に代入後，`append()` によりリスト化
- グラフ化に際しては，`numpy` 及び `matplotlib.pyplot` をインポートし
  - `xp = np.linspace(x[0],x[n-1],N+1)` で，x 座標の最小値から最大値を 400 分割したリストを生成
  - `plt.scatter(x,y,marker="o",c="red",s=100)` で予め与えた通過点のグラフを生成
  - `plt.plot(xp,z)` で，Lagrange 補間後のグラフを生成
- プログラムの実行では
  - 下記のようなインプットカードを，ファイル名 `in.txt` で用意し，
  - Linux シェルや Windows PowerShell のリダイレクト機能を使って，
    `$python3 ./lagrange.py < in.txt`
    で実行すると，キーボード入力を削減できる

```
7
1 0
1.5 -14.77
2 0
3 0
4 0
5 0
6 0
```

```
import numpy as np
import matplotlib.pyplot as plt

N=400

z = [0 for i in range(N+1)]
a = [1 for i in range(10)]
b = [1 for i in range(10)]

n = int(input())
x = []
y = []
for i in range(n):
    xx, yy = map(float, input().split(" "))
    x.append(xx)
    y.append(yy)
xstep = (x[n-1]-x[0])/N

for i in range(10):
    a[i]=1.0
    b[i]=1.0
for i in range(N+1):
    z[i] = 0.0
for i in range (n):
    for j in range(n):
        if i!=j:
            b[i] *= (x[i]-x[j])

xx = x[0]
for k in range(N+1):
    for i in range(n):
        a[i] = 1.0
        for j in range(n):
            if i!=j:
                a[i] *= (xx-x[j])
        z[k] += y[i]*a[i]/b[i]
    print(xx,z[k])
    xx += xstep

xp = np.linspace(x[0],x[n-1],N+1)
plt.scatter(x,y,marker="o",c="red",s=100)
plt.plot(xp,z)
plt.show()
```

2 課題の解答

- 赤丸は予め与えた通過点
- 青線は Lagrange 補間

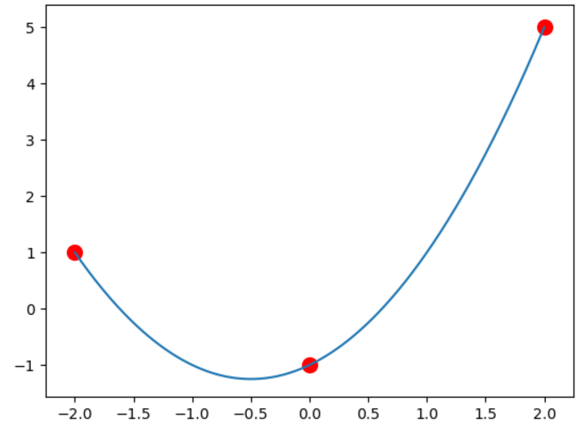

Figure 1: $(-2, 1), (0, -1), (2, 5)$ を通過するグラフ

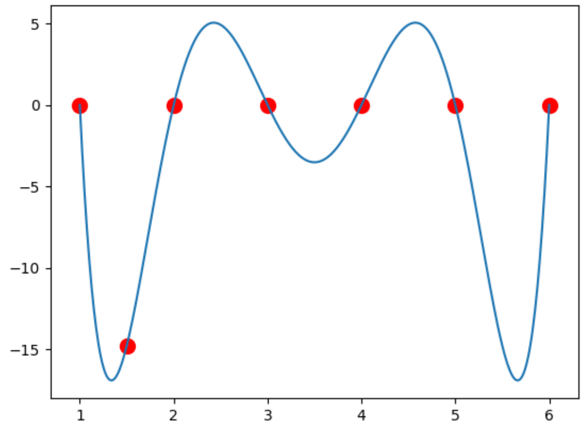

Figure 2: $(1, 0), (1.5, -14.77), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0)$ を通過するグラフ

3　Runge 現象

Runge 現象が見出した課題は，多項式補間の次数が大きくなると，振動が発生し，補間精度が著しく劣化することがある，ということである．この解決方法として，下記の式で示される間隔 (Chebychev's node) でサンプリングを行う方法がある．

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right), i = 1, 2, \cdots n$$

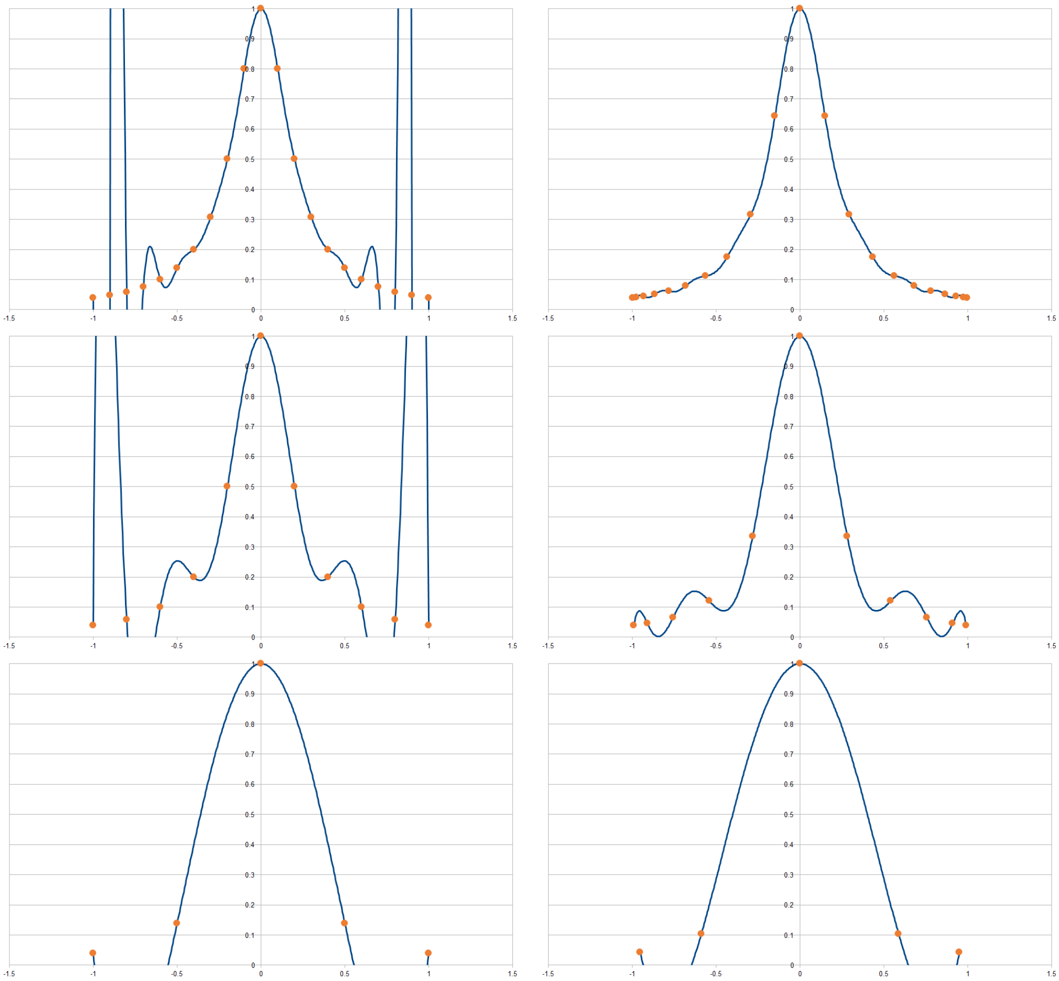

Figure 3: 左列：等間隔サンプリング，右列：Chebyshev's node サンプリング，上段：21 点サンプリング，中段：11 点サンプリング，下段：5 点サンプリング

4　Lagrange 補間の限界とその他の補間方式

Runge 現象で見た通り，Lagrange 補間はいつでも正しい補間になるわけではない．不等間隔で区間の端に行くほど超密なサンプリングにすること (Chebyshev's node) で回避することは可能であるが，そもそも実験的に得られるデータが Chebyshev's node のような分布を持っているとは限らず，その用途が限定的になる．

任意のサンプリング分布に対する補間としてスプライン補間がある．中でも B スプライン曲線による補間は，**コンピュータグラフィックスで多用**されている．ゲーム作成や CG 関係に興味がある —— 就職先として考えている人は，スプライン補間，及び B スプライン曲線について調べてみることを薦める．