Homework 2, Interpolation circle

資工 3B 00957144 蔣佳純

日期:2022/10/23

1. Generate N+1 sample points in a circle of radius equal to r (r=10.0)

N=8: (等間距)

	θ 0	θ 1	θ 2	θ 3	θ 4	θ 5	θ 6	θ 7	θ 8
θ[i] = i*Δθ	0π/ 4	1π/ 4	2π/ 4	3π/ 4	4π/ 4	5π/ 4	6π/ 4	7π/ 4	8π/ 4

	X 0	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8
X[i] = r*cos(θ[i])	10.00000	7.07107	0.00000	-7.07107	-10.00000	-7.07107	-0.00000	7.07107	10.00000

	Y 0	Y 1	Y 2	Y 3	Y 4	Y 5	Y 6	Y 7	Y 8
Y[i] = r*sin(θ[i])	0.00000	7.07107	10.00000	7.07107	0.00000	-7.07107	-10.00000	-7.07107	-0.00000

N=16: (等間距)

	θ 0	θ 1	θ 2	θ 3	θ 4	θ 5	θ 6	θ 7	θ 8
θ[i] = i*Δθ	0π/ 8	1π/ 8	2π/ 8	3π/ 8	4π/ 8	5π/ 8	6π/ 8	7π/ 8	8π/ 8

	θ 9	θ10	θ11	θ12	θ13	θ14	θ15	θ16
θ[i] = i*Δθ	9π/ 8	10π/ 8	11π/ 8	12π/ 8	13π/ 8	14π/ 8	15π/ 8	16π/ 8

	X 0	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8
X[i] = r*cos(θ[i])	10.00000	9.23880	7.07107	3.82683	0.00000	-3.82683	-7.07107	-9.23880	-10.00000

	X 9	X10	X11	X12	X13	X14	X15	X16
X[i] = r*cos(θ[i])	-9.23880	-7.07107	-3.82683	-0.00000	3.82683	7.07107	9.23880	10.00000

	Y 0	Y 1	Y 2	Y 3	Y 4	Y 5	Y 6	Y 7	Y 8
Y[i] = r*sin(θ[i])	0.00000	3.82683	7.07107	9.23880	10.00000	9.23880	7.07107	3.82683	0.00000

	Y 9	Y10	Y11	Y12	Y13	Y14	Y15	Y16
Y[i] = r*sin(θ[i])	-3.82683	-7.07107	-9.23880	-10.00000	-9.23880	-7.07107	-3.82683	-0.00000

N=5: (隨機取樣 1)

	θ 0	θ 1	θ 2	θ 3	θ 4	θ 5
θ[i] = i*Δθ	0	10◦	50◦	180◦	250◦	360◦(2π)

	X 0	X 1	X 2	X 3	X 4	X 5
X[i] = r*cos(θ[i])	10.00000	9.84808	6.42788	-10.00000	-3.42020	10.00000

	Y 0	Y 1	Y 2	Y 3	Y 4	Y 5
Y[i] = r*sin(θ[i])	0.00000	1.73648	7.66044	0.00000	-9.39693	-0.00000

N=5: (隨機取樣 2)

	θ0	θ1	θ2	θ3	θ4	θ5
θ[i] = i*Δθ	0	10°	50°	180°	340°	360°(2π)
	X0	X1	X2	X3	X4	X5
X[i] = r*cos(θ[i])	10.00000	9.84808	6.42788	-10.00000	9.39693	10.00000
	Y0	Y1	Y2	Y3	Y4	Y5
Y[i] = r*sin(θ[i])	0.00000	1.73648	7.66044	0.00000	-3.42020	-0.00000

N=10: (隨機取樣 3)

	θ0	θ1	θ2	θ3	θ4	θ5	θ6	θ7	θ8	θ9	θ10
θ[i] = i*Δθ	0	10°	30°	50°	100°	120°	180°	260°	300°	340°	360° (2π)
	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
X[i] = r*cos(θ[i])	10.00000	9.84808	8.66025	6.42788	-1.73648	-5.00000	-10.00000	-1.73648	5.00000	9.39693	10.00000
	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
Y[i] = r*sin(θ[i])	0.00000	1.73648	5.00000	7.66044	9.84808	8.66025	0.00000	-9.84808	-8.66025	-3.42020	-0.00000

N=8: (隨機取樣 4)

	θ0	θ1	θ2	θ3	θ4	θ5	θ6	θ7	θ8
θ[i] = i*Δθ	0	10°	30°	50°	100°	120°	180°	200°	360°(2π)
	X0	X1	X2	X3	X4	X5	X6	X7	X8
X[i] = r*cos(θ[i])	10.00000	9.84808	8.66025	6.42788	-1.73648	-5.00000	-10.00000	-9.39693	10.00000
	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
Y[i] = r*sin(θ[i])	0.00000	1.73648	5.00000	7.66044	9.84808	8.66025	0.00000	-3.42020	-0.00000

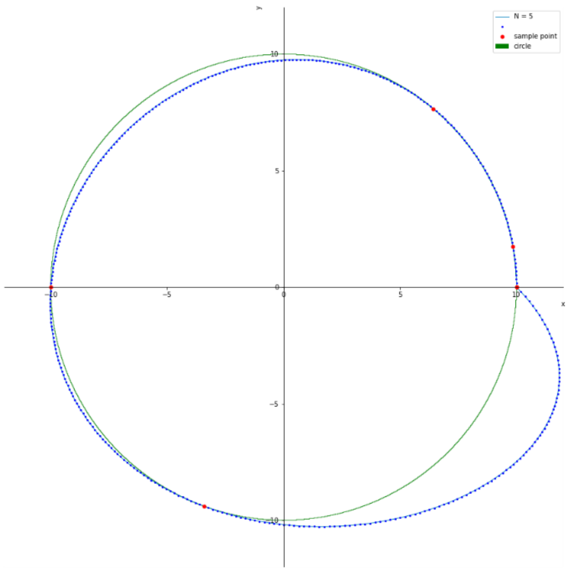
2. Produce 361 points in the xy-plane by using the following two methods

A. $\Delta t = \frac{2\pi}{360}$, px[i] = r $*$ cos(i $*$ Δt) , py[i] = r $*$sin(i $*$ Δt), $0 \le i \le 360$.

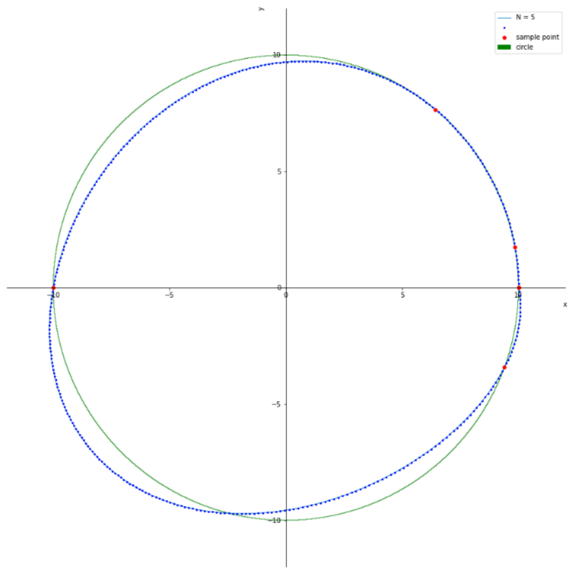
B. Using Newton's polynomial and keep the interpolation points in qx[i] and qy[i]

※綠色線為圓函數圖，藍色點為內插點(interpolation points)，紅色點為取樣點(sample points)

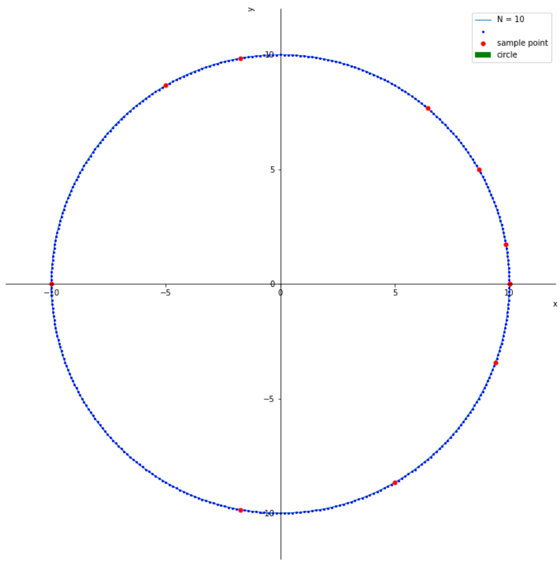
N=5: (隨機取樣 1)中的結果圖：



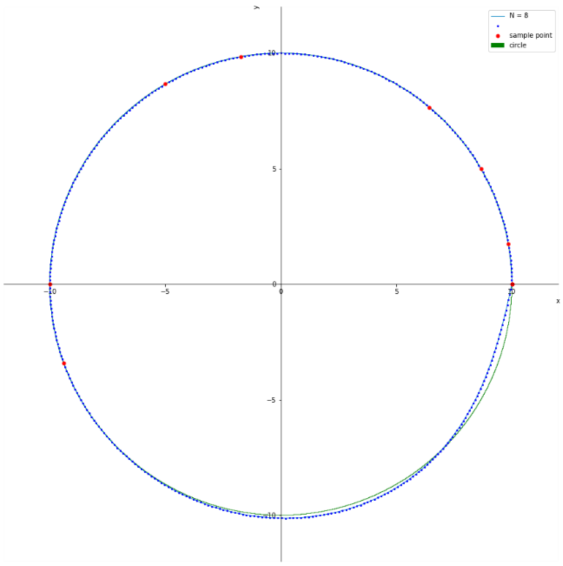
N=5: (隨機取樣 2)中的結果圖：



N=10: (隨機取樣 3)中的結果圖：

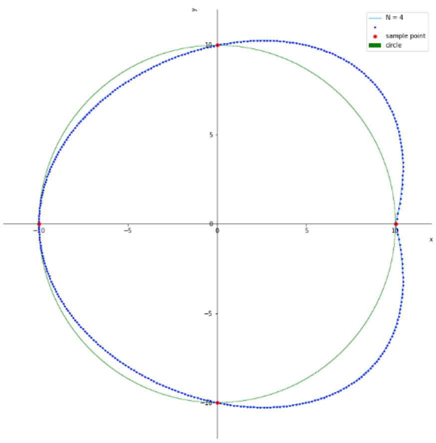


N=8: (隨機取樣 4)中的結果圖：

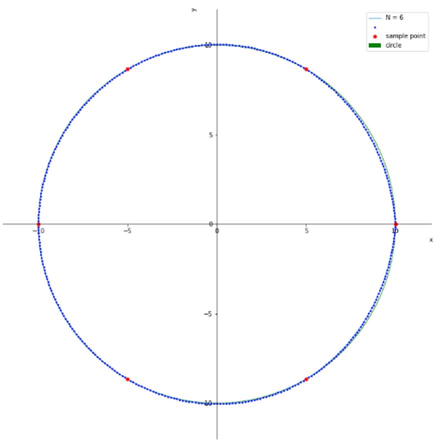


等間距取點結果圖：

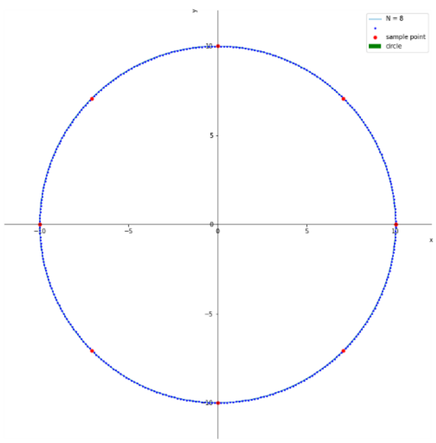
N = 4



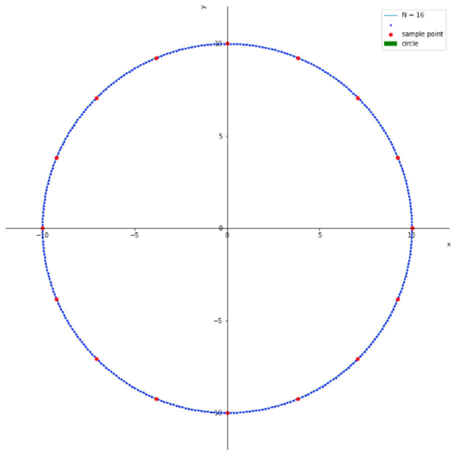
N = 6

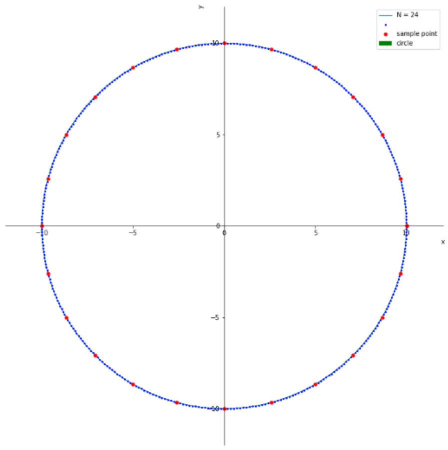


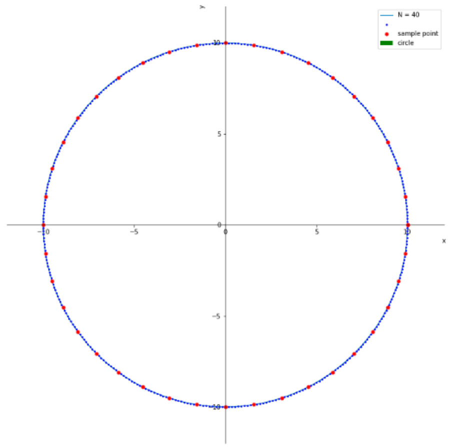
N = 8



N = 16 N = 24 N = 40







3. Compute the accumulative distance between the interpolation points calculated by these two methods:

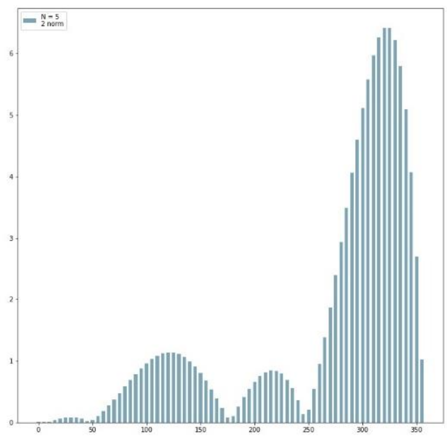
   A. 2norm = $\sqrt{\sum_{i=0}^{359}[(px[i]-qx[i])^2+(py[i]-qy[i])^2]}$

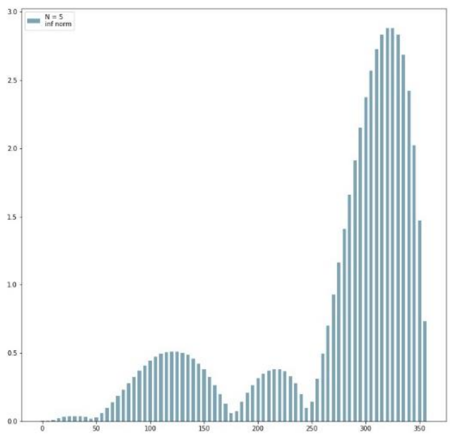
   B. ∞ norm = $\max\limits_{0\le i\le 359}\sqrt{[(px[i]-qx[i])^2+(py[i]-qy[i])^2}$

```
ld error_2() {// 2 norm
	ld sum = 0;
	for (int i = 0; i < 360; i++)
		sum += ((px[i] - qx[i]) * (px[i] - qx[i]) + (py[i] - qy[i]) * (py[i] - qy[i]));
	return sqrt(sum);
}
ld error_inf() {// inf norm
	ld sum = 0;
	for (int i = 0; i < 360; i++)
		sum = max(sum, sqrt((px[i] - qx[i]) * (px[i] - qx[i]) + (py[i] - qy[i]) * (py[i] - qy[i])));
	return sum;
}
```

4. 誤差分析

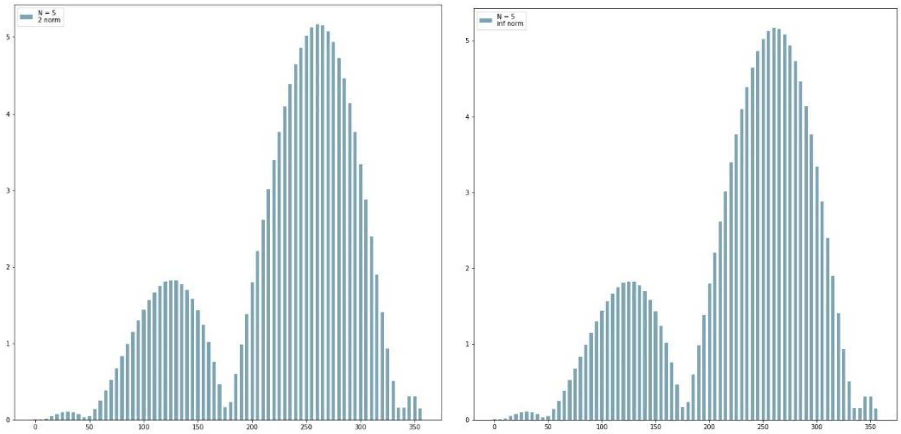
   N=5: (隨機取樣 1)中的結果圖：





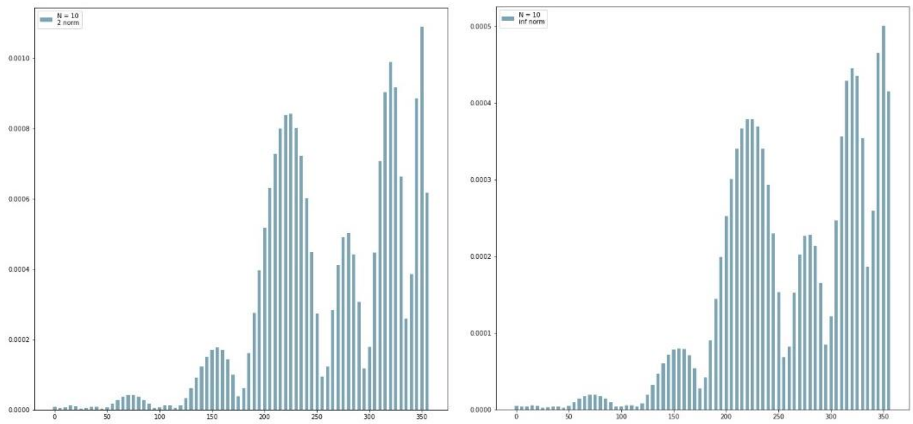
```
2 norm: 20.71840506803200199215  ∞ norm: 2.88280875317205964592
```

N=5: (隨機取樣 2)中的結果圖：



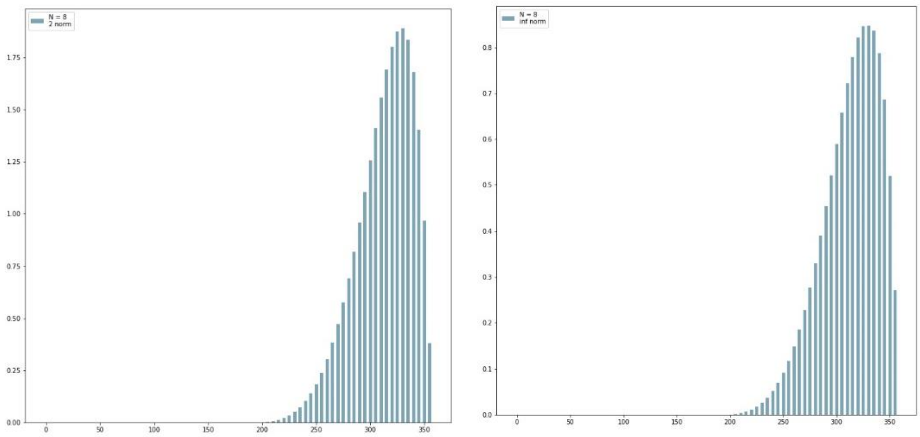
2 norm: 20.858341165375069437OO ∞ norm: 2.3151137943482762438O

N=10: (隨機取樣 3)中的結果圖：



2 norm: 0.00359969828754554654 ∞ norm: 0.00050090673282823512

N=8: (隨機取樣 4)中的結果圖：



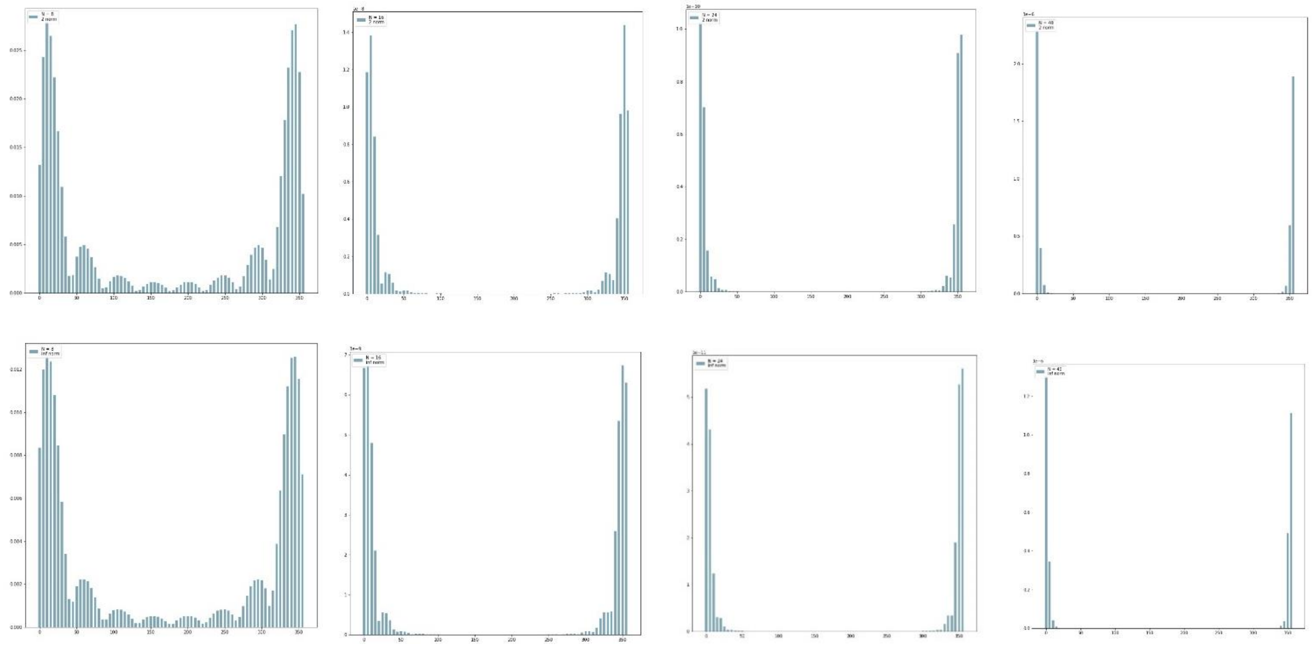
2 norm: 5.71374188390470916232 ∞ norm: 0.84753198198009682685

每兩個樣本點之間，離樣本點遠的地方的誤差會比靠近樣本點的誤差還要小。(2 norm 或是∞ norm 都是)

在隨機取樣本點的圖，會發現並不是在 end intervals 就一定會有最大誤差。雖然在 end intervals 的誤差有比較大，但在兩個樣本點間距很大的地方誤差也會變大。

等間距取點結果圖：

2 norm: 0.08125623303642762196	2 norm: 0.00000002884120514204	2 norm: 0.00000000018510216434	2 norm: 0.00000305695707355610
∞ norm: 0.01258927570960070678	∞ norm: 0.00000000674464550049	∞ norm: 0.00000000005616550554	∞ norm: 0.00000130235315916593
N=8	N=16	N=24	N=40



※ 上面四張是 2 norm，下面四張是 ∞ norm (每五個點計算一次)

由圖可看出，不管 N 為多少，也不論是 2 norm 或是∞ norm，在 end intervals 有最大誤差，在 middle intervals 則有最小誤差。

並非 N 越大誤差越小，N=40 時誤差比 N=24 大(縱坐標單位不同)，但 N 越大有誤差的點有變少的趨勢。

發現每兩個樣本點之間，離樣本點遠的地方的誤差會比靠近樣本點的誤差還要小。

結論：

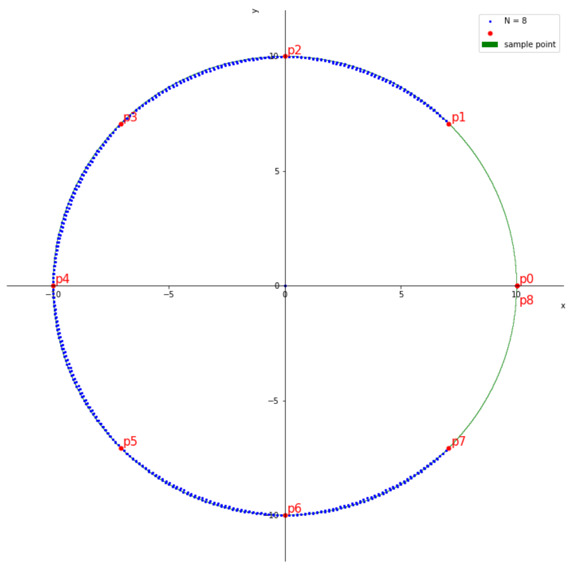
在兩個樣本點間距很大的地方有比較大的誤差

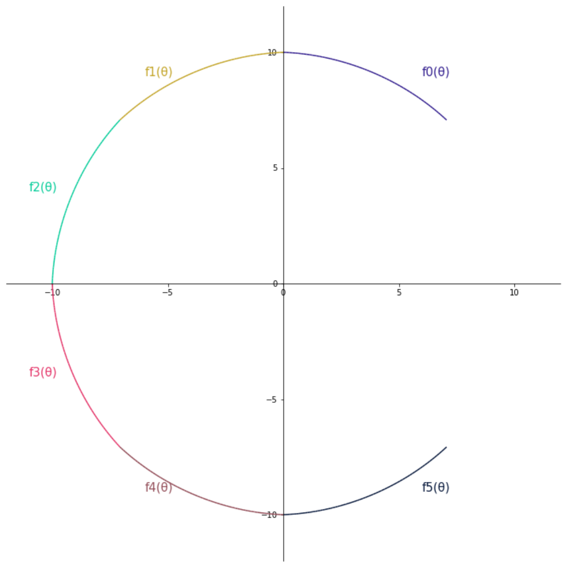
end intervals 也會有比較大的誤差

並不是 end intervals 就一定有最大誤差

在某兩個樣本點之間，離樣本點遠的地方的誤差會比靠近樣本點的誤差還要小

5. piecewise cubic spline method (N=8，分 8 個區域)

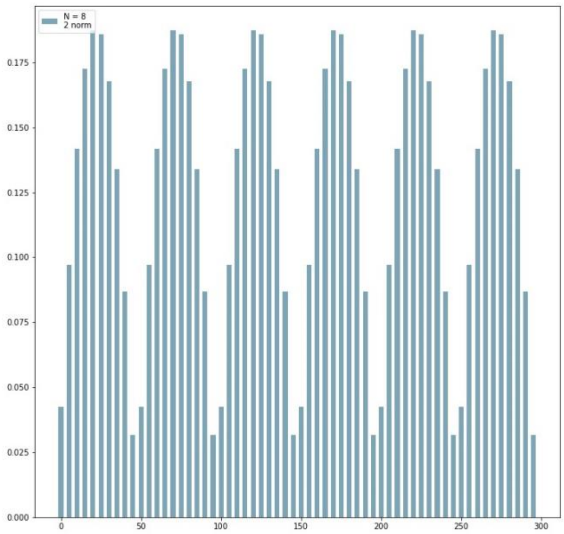


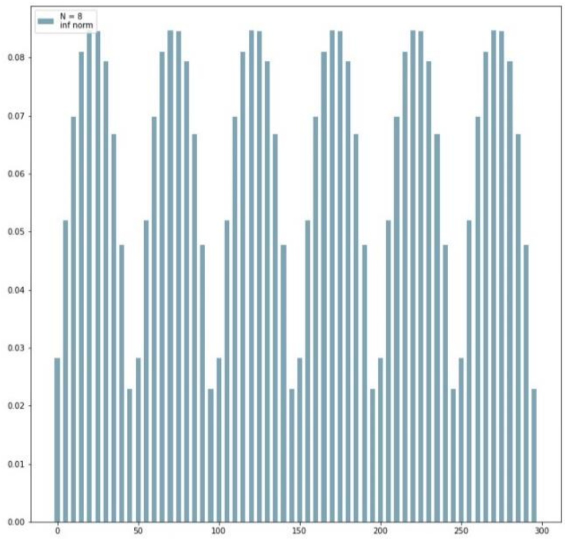


左圖是點代入每個區域的函數後繪出點點
右圖是分段繪出每個區域的函數圖
雖然不是很明顯，但能看出相接處有點不平滑。
f0: p0,p1,p2,p3 為取樣點
f1: p1,p2,p3,p4 為取樣點
f2: p2,p3,p4,p5 為取樣點
f3: p3,p4,p5,p6 為取樣點
f4: p4,p5,p6,p7 為取樣點
f5: p5,p6,p7,p8 為取樣點

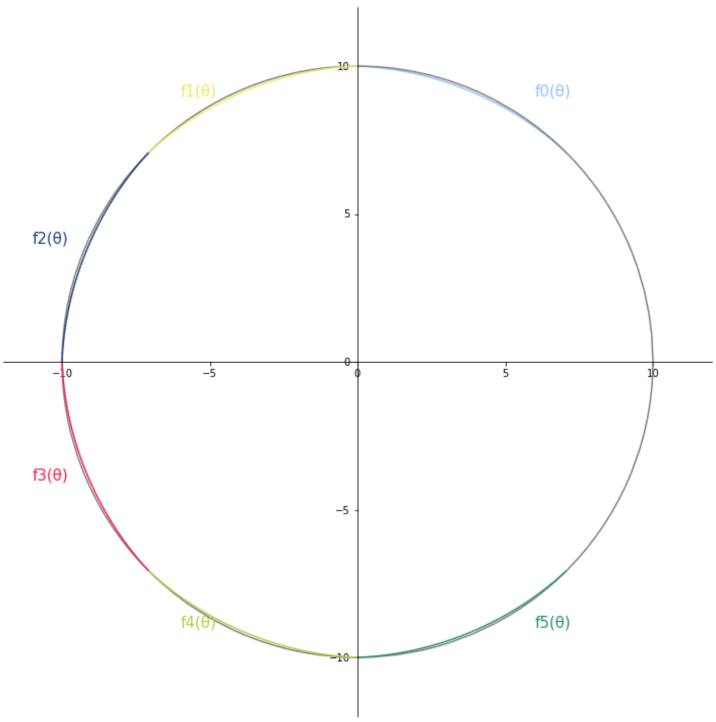
6. Compute the errors. Compare the results with those of the Newton polynomial.

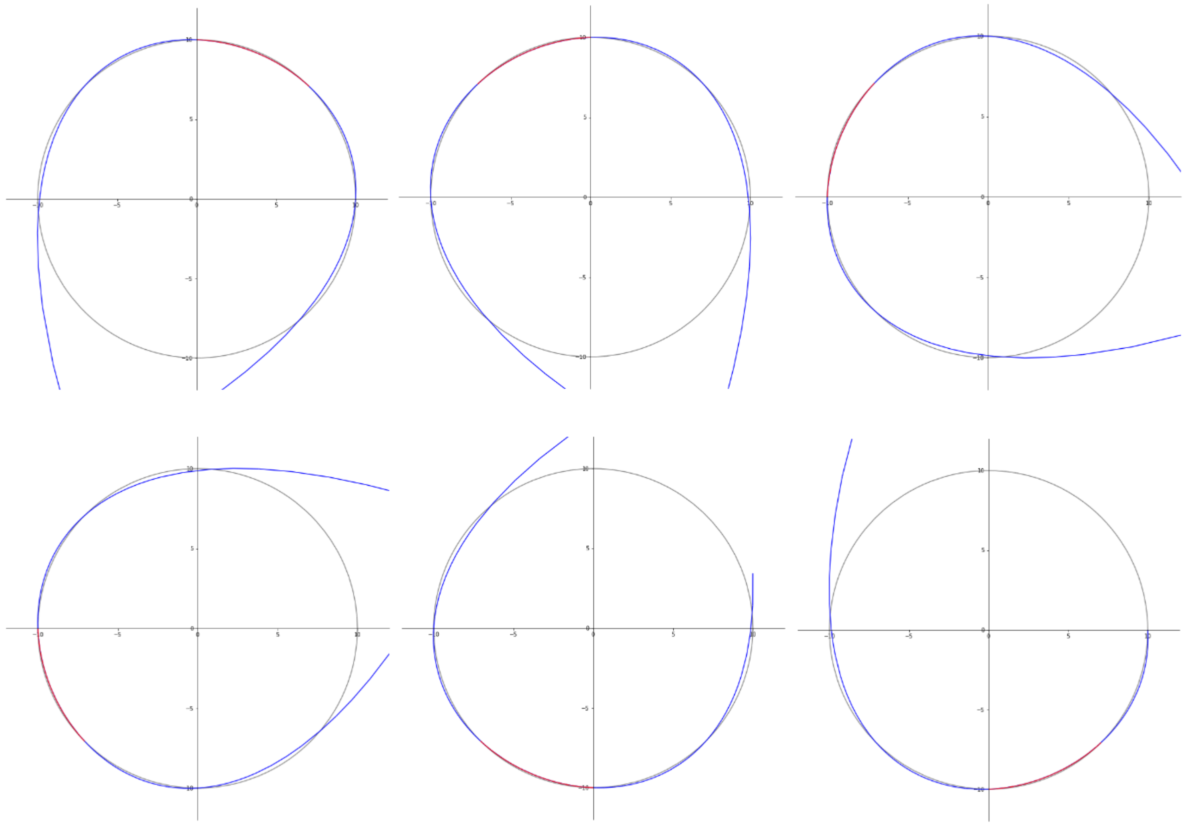
2 norm: 1.0543742196007219562
∞ norm: 0.08470954970438765441





piecewise cubic splines 的誤差比 Newton’s method 的還要大一些(同為等距 N=8)
和牛頓法一樣，每兩個樣本點之間，中間的誤差會比靠近樣本點的誤差還要大。(因為這也是牛頓多項式)





※每區的 3 階多項式拿來繪出全範圍(藍色)，以及真正拿去做內插的範圍(紅色)

可以看出，紅色區域仍然是整個區域中誤差最小的(因為仍是用牛頓多項式)。

7. Are the piecewise cubic splines enjoy C1-continuity at the sample points?

接合點為 p1~p7：

p1: 以 f0,f1 作為連接的兩函數

p2: 以 f0,f2 作為連接的兩函數

p3: 以 f0,f3 作為連接的兩函數

p4: 以 f1,f4 作為連接的兩函數

p5: 以 f2,f5 作為連接的兩函數

p6: 以 f3,f5 作為連接的兩函數

p7: 以 f4,f5 作為連接的兩函數

連接 p1~p7 的兩個函數的微分值：

p1:

X'(θ1) = -7.245186202974 Y'(θ1) = 6.730286672971

X'(θ1) = -7.245186202974 Y'(θ1) = 8.186642470151

p2:

X'(θ2) = -9.882151640869 Y'(θ2) = 0.364088949295

X'(θ2) = -10.911950700876 Y'(θ2) = 0.665710110712

p3:

X'(θ3) = -7.245186202974 Y'(θ3) = -8.186642470151

X'(θ3) = -8.186642470151 Y'(θ3) = -7.245186202974

p4:

X'(θ4) = 0.665710110712 Y'(θ4) = -10.911950700876

X'(θ4) = -0.665710110712 Y'(θ4) = -10.911950700876

```
p5:
X'(θ5) = 8.186642470151 Y'(θ5) = -7.245186202974
X'(θ5) = 7.245186202974 Y'(θ5) = -8.186642470151
p6:
X'(θ6) = 10.911950700876 Y'(θ6) = 0.665710110712
X'(θ6) = 9.882151640869 Y'(θ6) = 0.364088949295
p7:
X'(θ7) = 7.245186202974 Y'(θ7) = 8.186642470151
X'(θ7) = 7.245186202974 Y'(θ7) = 6.730286672971
```

※可見微分值不同，方向也不同，因此不為 C1- continuity。

連接 p1~p7 的兩個函數值：

```
p1:
X(θ1) = 7.071067811865 Y(θ1) = 7.071067811865
X(θ1) = 7.071067811865 Y(θ1) = 7.071067811865
p2:
X(θ2) = 0.000000000000 Y(θ2) = 10.000000000000
X(θ2) = 0.000000000000 Y(θ2) = 10.000000000000
p3:
X(θ3) = -7.071067811865 Y(θ3) = 7.071067811865
X(θ3) = -7.071067811865 Y(θ3) = 7.071067811865
p4:
X(θ4) = -10.000000000000 Y(θ4) = 0.000000000000
X(θ4) = -10.000000000000 Y(θ4) = 0.000000000000
p5:
X(θ5) = -7.071067811865 Y(θ5) = -7.071067811865
X(θ5) = -7.071067811865 Y(θ5) = -7.071067811865
p6:
X(θ6) = -0.000000000000 Y(θ6) = -10.000000000000
X(θ6) = -0.000000000000 Y(θ6) = -10.000000000000
p7:
X(θ7) = 7.071067811865 Y(θ7) = -7.071067811865
X(θ7) = 7.071067811865 Y(θ7) = -7.071067811865
```

※函數值相同，因此為 C0- continuity。

8. 想法和心得

使用牛頓法若是取樣本點時取的很糟糕(間隔差太多)，就有可能得到很大的誤差。

但在等距取樣本點時，牛頓法表現比預想中的還要好，同樣是等距 N=8，牛頓法的誤差比 peicewise cubic splines 更小。牛頓法在 N=6 時還稍微能看出誤差，但 N=8 之後看起來都差不多，誤差很小。

課本上的例子使用 peicewise cubic splines 時用的函數不是用牛頓內插創造出的，且課本上的例子是 C1- continuity，可能是因為建構函數方式不同，才導致最後結果不為 C1- continuity。