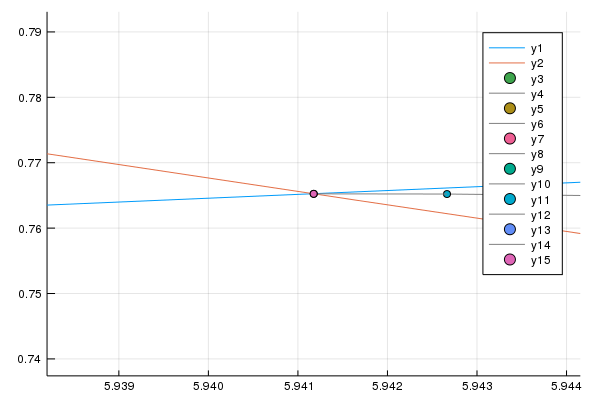
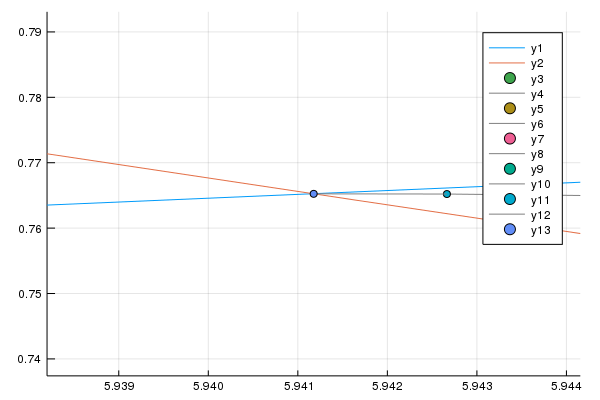
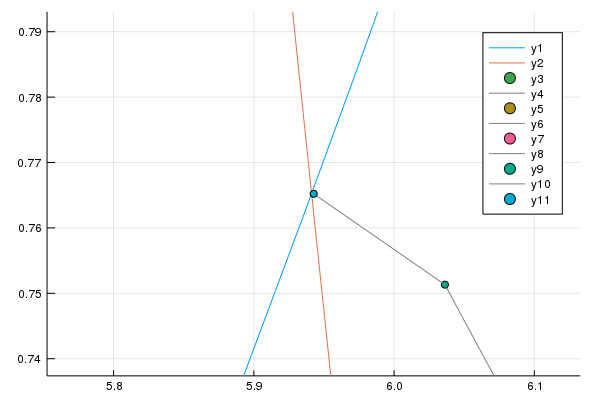
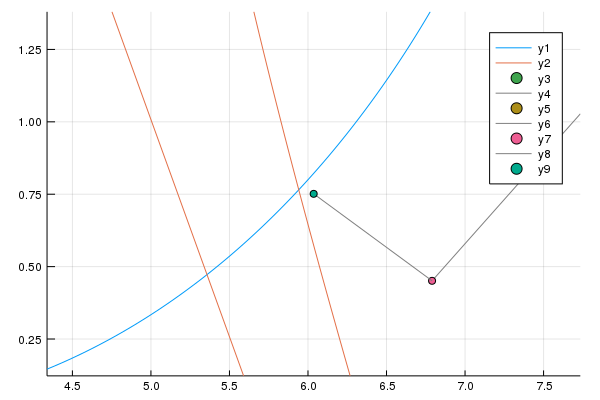
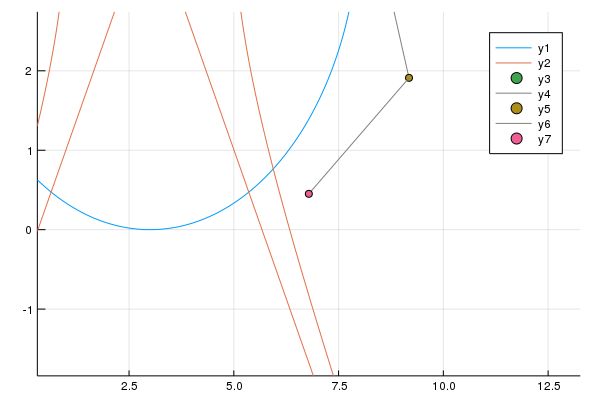
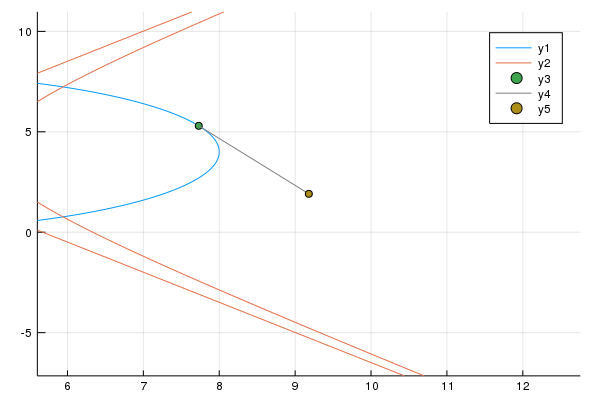
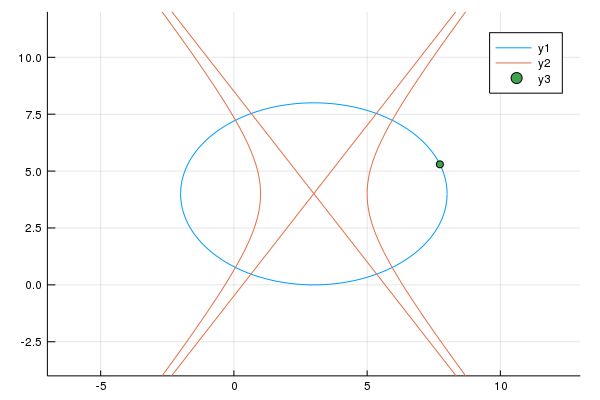
數值分析

作業一

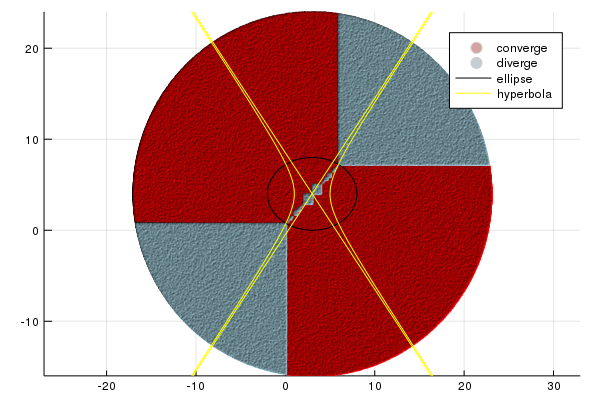
指導老師: 翁世光 老師

學生: 黃楚祐 學號: 00557043

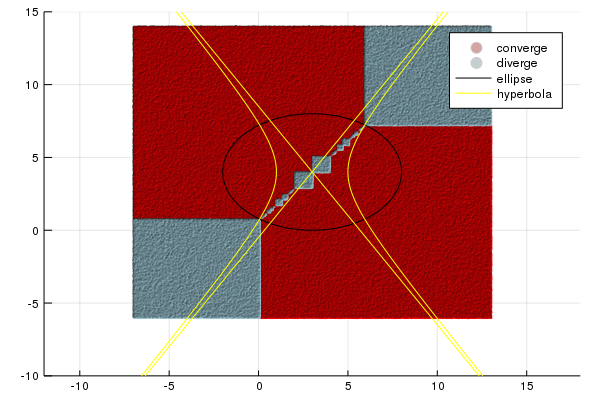
1. 以在橢圓上面隨機的點為例 : 做六次修正就可以收斂到誤差範圍小於10-6



1. 隨機取點106次，範圍限制在以(3, 4) 為圓心，半徑為20的範圍



1. 隨機取點106次，更小一點的矩形，放大1.5倍檢視



1. 由2、3可見在同時大於上角交點的X和Y會發散;同時小於左下角交點的 X和Y也會發散。
2. 收斂結果輸出(10筆測資):

i xn yn error

------------------------------------------------------------------------

0 3.010000 3.990000 ------

1 435.530952 -519.188391 678.815146

2 219.275476 -257.604195 339.400534

3 111.157737 -126.822097 169.686189

4 57.118859 -61.451040 84.814948

5 30.139351 -28.805455 42.351246

6 16.729048 -12.562208 21.063696

7 10.179568 -4.596994 10.312144

8 7.192224 -0.907062 4.747612

9 6.127846 0.480284 1.748609

10 5.946747 0.753706 0.327958

11 5.941182 0.765220 0.012789

12 5.941176 0.765241 0.000021

13 5.941176 0.765241 0.000000

------------------------------------------------------------------------

Root= (5.941176, 0.765241, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 2.708090 7.993177 ------

1 -11.963077 7.306782 14.687216

2 -4.770601 7.235544 7.192829

3 -1.441919 7.234759 3.328682

4 -0.194696 7.234759 1.247223

5 0.048764 7.234759 0.243460

6 0.058806 7.234759 0.010042

7 0.058824 7.234759 0.000017

8 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

i xn yn error

------------------------------------------------------------------------

0 -0.504898 6.852725 ------

1 0.013490 7.260340 0.659451

2 0.058479 7.234860 0.051704

3 0.058824 7.234759 0.000358

4 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 6.885464 6.517544 ------

1 7.715006 5.698166 1.165985

2 9.155170 3.466377 2.656117

3 6.780289 -6.071174 9.828781

4 6.034305 -1.555073 4.577298

5 5.942606 0.280652 1.838013

6 5.941177 0.733672 0.453023

7 5.941176 0.765088 0.031416

8 5.941176 0.765241 0.000153

9 5.941176 0.765241 0.000000

------------------------------------------------------------------------

Root= (5.941176, 0.765241, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 -1.453719 5.818024 ------

1 -0.198016 7.786771 2.335114

2 0.048510 7.274994 0.568059

3 0.058806 7.235006 0.041291

4 0.058824 7.234759 0.000248

5 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

i xn yn error

------------------------------------------------------------------------

0 5.813349 7.306727 ------

1 5.682617 7.377912 0.148856

2 5.411597 7.518031 0.305098

3 4.823871 7.789898 0.647560

4 3.364335 8.304378 1.547558

5 -8.325145 9.241099 11.726952

6 -3.044489 7.618782 5.524241

7 -0.737815 7.255135 2.335162

8 -0.026070 7.234823 0.712035

9 0.057633 7.234759 0.083703

10 0.058823 7.234759 0.001191

11 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 1.025910 7.675037 ------

1 -0.178059 7.261132 1.273130

2 0.049995 7.234866 0.229562

3 0.058810 7.234759 0.008816

4 0.058824 7.234759 0.000013

5 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 -1.945982 4.586384 ------

1 -0.347491 13.215389 8.775814

2 0.034165 9.175423 4.057954

3 0.058721 7.598611 1.577003

4 0.058824 7.253154 0.345458

5 0.058824 7.234811 0.018342

6 0.058824 7.234759 0.000052

7 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 2.852748 7.998265 ------

1 -26.446844 7.307659 29.307730

2 -11.870306 7.235563 14.576717

3 -4.726018 7.234759 7.144287

4 -1.422840 7.234759 3.303179

5 -0.189357 7.234759 1.233483

6 0.049167 7.234759 0.238524

7 0.058808 7.234759 0.009640

8 0.058824 7.234759 0.000016

9 0.058824 7.234759 0.000000

------------------------------------------------------------------------

Root= (0.058824, 7.234759, error=0.000000)

------------------------------------------------------------------------

i xn yn error

------------------------------------------------------------------------

0 4.162297 7.890424 ------

1 1.022144 8.490839 3.197039

2 -0.175771 7.410421 1.613166

3 0.050159 7.239283 0.283429

4 0.058811 7.234762 0.009762

5 0.058824 7.234759 0.000013

6 0.058824 7.234759 0.000000

Root= (0.058824, 7.234759, error=0.000000)

1. 我這次在2D Newton’s Method設定在做64次修正以內收斂完畢的點才算收斂，不太確定這樣是不是夠多次了，但是我覺得就上面的圖2、圖3來看還滿賞心悅目，也可以觀察出一些規則。
2. 我猜其中的一些點會被判為發散可能是因為一開始修正的值太大了，所以溢位後導致點跑到不能收斂的地方。
3. 我覺得這個找根方法真的滿快的，常常一下子逼近兩三位數，缺點是發散的情形出現的頻率也不低、在這個函數的四個根也只會收斂到左上和右下的根。