1.

(1) $ax_1^2+2bx_1x_2+cx_2^2+dx_1+ex_2=1$

$$[x_1\ x_2]\begin{bmatrix}a & b\\ b & c\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+[d\ e]\begin{bmatrix}x_1\\ x_2\end{bmatrix}=1$$

$$\therefore A=\begin{bmatrix}a & b\\ b & c\end{bmatrix}、B=[d、e]_{\#}$$

(2) circle : $x_1^2+x_2^2=r^2 \Rightarrow \frac{1}{r^2}x_1^2+\frac{1}{r^2}x_2^2=1$

$a=c=\frac{1}{r^2}$, $b=d=e=0$

$$A=\begin{bmatrix}\frac{1}{r^2} & 0\\ 0 & \frac{1}{r^2}\end{bmatrix}_{\#}$$

parabola : $x_2^2=4\alpha x_1 \Rightarrow x_2^2-4\alpha x_1=0$

$a=0$、$b=0$、$c=1$、$d=-4\alpha$、$e=0$

$$A=\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}_{\#}$$

ellipse : $\frac{1}{\alpha^2}x_1^2+\frac{1}{\beta^2}x_2^2=1$

$a=\frac{1}{\alpha^2}$, $b=d=e=0$, $c=\frac{1}{\beta^2}$

$$A=\begin{bmatrix}\frac{1}{\alpha^2} & 0\\ 0 & \frac{1}{\beta^2}\end{bmatrix}_{\#}$$

hyperbola : $\frac{1}{\alpha^2}x_1^2-\frac{1}{\beta^2}x_2^2=1$

$a=\frac{1}{\alpha^2}$, $c=-\frac{1}{\beta^2}$, $b=d=e=0$

$$A=\begin{bmatrix}\frac{1}{\alpha^2} & 0\\ 0 & -\frac{1}{\beta^2}\end{bmatrix}_{\#}$$

2.

解 system 2~6 就是解這個方程組，可以得到中心點 z,w。

$$\begin{bmatrix}-2a & -b\\ -b & -2c\end{bmatrix}\begin{bmatrix}z\\ w\end{bmatrix}=\begin{bmatrix}d\\ e\end{bmatrix}.$$

接下來解下圖的線性方程組可以得出 a’, b’, c’，就可以有方程式 $a'(x')^2+b'x'y'+c'(y')^2=1$，再把它寫成 $x^TAx=1$ 的樣子，A 即是由 a’, b’, c’組成。利用 numpy.linalg.eig 得出 eigen values, vectors 的回傳值，根據課本 6.6，alpha、beta 即為 eigen values 的倒數，旋轉矩陣 U 則為 eigen vectors 的 transpose。

$$\begin{bmatrix}z^2+1/a & zw & w^2\\ z^2 & zw+1/b & w^2\\ z^2 & zw & w^2+1/c\end{bmatrix}\begin{bmatrix}a'\\ b'\\ c'\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 1\end{bmatrix}.$$

3.

$$S=\frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix}(x_i-\bar{x})^2 & (x_i-\bar{x})(y_i-\bar{y})\\ (x_i-\bar{x})(y_i-\bar{y}) & (y_i-\bar{y})^2\end{bmatrix}.$$

我先算出 S 是長甚麼樣子，把 Σ 丟進矩陣裡就可以知道算出的總和矩陣左上角

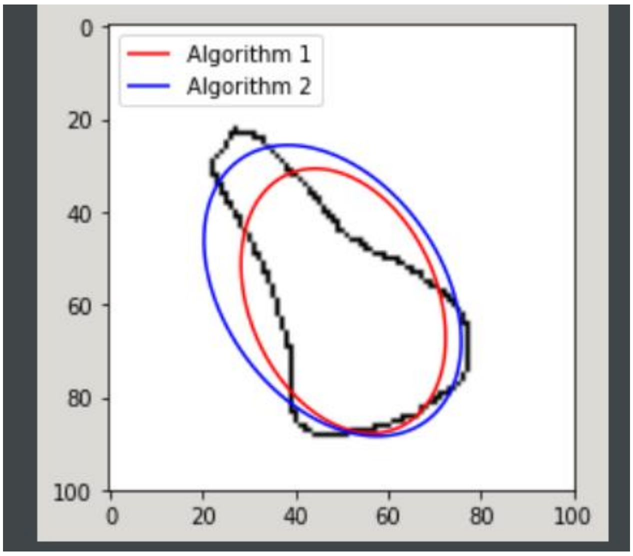
為 $\Sigma(x_i\text{-x bar})^2$、右上與左下:$\Sigma(x_i\text{-x bar})(y_i\text{-y bar})$、右下:$\Sigma(y_i\text{-y bar})^2$。

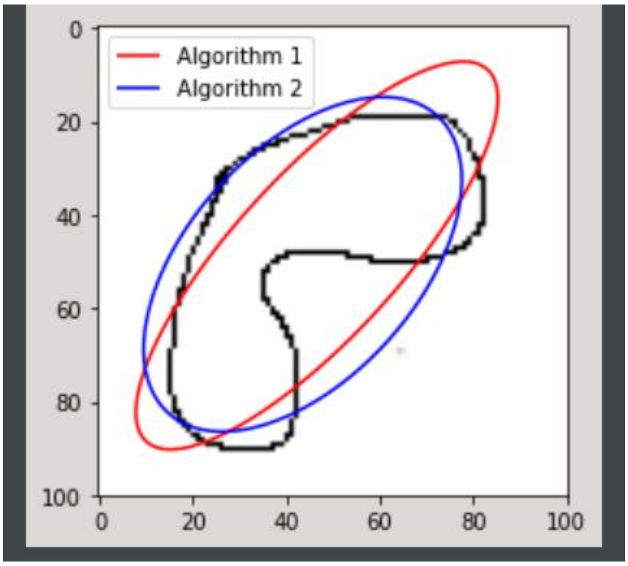
```
# compute the covariance matrix Sigma, remember to sum it up
ai_square = np.sum(np.array( [ai**2 for ai in Y[:,0]] ))    #Y[:,0] = xi - average of x
ai_dot_bi = np.sum(np.array( [pair[0]*pair[1] for pair in Y] ))
bi_square = np.sum(np.array( [bi**2 for bi in Y[:,1]] ))    #Y[:,1] = yi - average of y

sigma = (1/n) * np.array([ [ai_square, ai_dot_bi],
                           [ai_dot_bi, bi_square]])
```

再利用 np.sum()就可以算出這 4 個元素，進而形成 sigma 矩陣，最後求 sigma 矩陣的 eigen vectors 即為 U 之旋轉矩陣。

4.





上面兩張都是自己畫的圖形，我發現如果圖形某處凹陷地特別明顯，那麼 algo2 普遍會比 algo1 還更像原圖，且凹陷程度越大，兩種演算法預測出來的圖形相差也越大。

5.

(1)The rank of A = the number of nonzero singular values。

(2)可以從 singular value 的大小知道 matrix A 跟 rank 較低的矩陣之間，它們的 Frobenius norm 有多接近。(下圖截自上課講義 p.83/92)

If A is an $m \times n$ matrix with rank r and $0 < k < r$, we can use the <u>singular value decomposition</u> to find a matrix in $R^{m \times n}$ of rank k that is closest to A with respect to the <u>Frobenius norm</u>