1. a. This method converges in 19 iterations to the approximate root [0.50000038 0.86602519] as shown in my Python code pushed to Git.

   b. This matrix is a good choice because it’s the inverse of the Jacobian at the initial guess scaled by a factor of ⅙ to reduce the step size.

   c. Newton’s method converged to the approximate root [0.50000007 0.86602553] in 12 iterations as shown in my Python code pushed to Git.

   d. Since both methods were used with a tolerance of 10^-6, the root is approximately [0.5, 0.866025]

1 d. $f(0.5, 0.866025) = 3(0.5)^2 - 0.866025^2 = 0.75 - 0.75 = 0$

$g(0.5, 0.866025) = 3(0.5)(0.866025)^2 - 0.5^3 - 1$

$= 1.125 - 0.125 - 1 = 0$

2.

Functions are continuous when $2/3 \leq \sqrt{1+(x_n+y_n)^2}$

this is true for all x_n and y_n.

$$\frac{\partial G}{\partial x} = \frac{-2(x+y)}{3\sqrt{2(1+(x+y)^2)}}$$

$$\frac{\partial G}{\partial y} = \frac{-2(x+y)}{3\sqrt{2(1+(x+y)^2)}}$$

Minimums of $\frac{\partial G}{\partial x}$ and $\frac{\partial G}{\partial y}$ occur when $x+y=0$

D includes all values of (x,y) where $x+y=0$

3\. a. $f(x, y) = 0$

$f/(f_x^2 + f_y^2) = [x_n + y_n] \Rightarrow f/(f_x^2 + f_y^2) - x_n - y_n = 0$

Newton: $\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix} \begin{bmatrix} f(x,y) \\ f/(f_x^2 + f_y^2) - x_n - y_n \end{bmatrix}$

$$\boxed{\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n - d f_x \\ y_n - d f_y \end{bmatrix}}$$

b. Starting at [1, 1, 1], This converges quadratically to the root [0, 1, 1] because the best fit of the table of successive errors $e_{n+1} = k|e_n|^p$ when the value of p is 2.