APPM 4600 — HOMEWORK # 6

1. The nonlinear system

$$\begin{cases} f(x,y) = x^2 + y^2 - 4 = 0 \\ g(x,y) = e^x + y - 1 = 0 \end{cases}$$

has two real solutions. In the last homework you used Newton's method to solve this problem with following initial guesses:

(i) $x = 1$, $y = 1$

(ii) $x = 1$, $y = -1$

(iii) $x = 0$, $y = 0$

In this assignment, use the two quasi-Newton methods with the different initial guesses.

Is the performance better or worse that of Newton's methods?

Soln: The code for this problem is below. It is a slight modification of the code posted on the canvas page.

For the initial guess of $\boldsymbol{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, Lazy Newton does not converge while Broyden does converge in 14 iterations. Lazy Newton does not converge because the largest eigenvalue of the inverse of the Jacobian is larger than 1.

For the initial guess of $\boldsymbol{x}_0 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, both Lazy Newton and Broyden converge. Lazy Newton takes 34 iterations while Broyden only takes 8.

For the initial guess of $\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, none of the iterations converge because the Jacobian of the initial guess is singular.

When Newton converges, it converges faster than the other methods.

2. Consider the nonlinear system

$$\begin{aligned} x + \cos(xyz) - 1 &= 0, \\ (1-x)^{1/4} + y + 0.05z^2 - 0.15z - 1 &= 0, \\ -x^2 - 0.1y^2 + 0.01y + z - 1 &= 0. \end{aligned}$$

Using your own codes test the following three techniques for approximating the solution to the nonlinear system to within 10^{-6}:

- Newton's method
- Steepest descent method
- First Steepest descent method with a stopping tolerance of 5×10^{-2}. Use the result of this as the initial guess for Newton's method.

Using the same initial guess, which technique converges the fastest? Try to explain the performance.

Soln:

With an initial guess of $(0, 0, 0)$, Newton converges in 4 iterations while steepest descent requires 6. The hybrid of using steepest descent to start the iteration only reduces the iteration count for Newton by 1.

For this problem, it looks like Newton is hard to beat.