

## Homework6

Sunday, October 8, 2023 11:04 PM

Tony Samour  
APPM 4600

1.) 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 than of Newton's methods?

Goal: Compare the 3 methods using the given initial guesses.

For use in the Jacobian:

$$\begin{cases} f_x(x, y) = 2x \\ f_y(x, y) = 2y \\ g_x(x, y) = e^x \\ g_y(x, y) = 1 \end{cases} \Rightarrow J = \begin{bmatrix} 2x & 2y \\ e^x & 1 \end{bmatrix}$$

Output:

```

C:\Users\tonys\Documents\APPM4600\Samour_APPM4600\Homework\Homework6\Problem1.py
C:\Users\tonys\Documents\APPM4600\Samour_APPM4600\Homework\Homework6\Problem1.py
File "C:\Users\tonys\Documents\APPM4600\Samour_APPM4600\Homework\Homework6\Problem1.py", line 109: RuntimeWarning: overflow encountered in exp
  F[1] = np.exp(X[0]) + X[1] - 1
Initial Guess: (1,1)
[-1.81626407  0.8373678 ]
Newton: the error message reads: 0
Newton: took this many seconds: 0.0006199216842651367
Newton: number of iterations is: 7
[nan nan]
Lazy Newton: the error message reads: 1
Lazy Newton: took this many seconds: 0.0
Lazy Newton: number of iterations is: 99
[-1.81626407  0.8373678 ]
Broyden: the error message reads: 0
Broyden: took this many seconds: 0.0
Broyden: number of iterations is: 12

Initial Guess: (1,-1)
[ 1.00416874 -1.72963729]
Newton: the error message reads: 0
Newton: took this many seconds: 0.00031257152557373046
Newton: number of iterations is: 5
[ 1.00416874 -1.72963729]
Lazy Newton: the error message reads: 0
Lazy Newton: took this many seconds: 0.0
Lazy Newton: number of iterations is: 36
[ 1.00416874 -1.72963729]
Broyden: the error message reads: 0
Broyden: took this many seconds: 0.0
Broyden: number of iterations is: 6
Traceback (most recent call last):
  File "C:\Users\tonys\Documents\APPM4600\Samour_APPM4600\Homework\Homework6\Problem1.py", line 225, in <module>
    driver()
  File "C:\Users\tonys\Documents\APPM4600\Samour_APPM4600\Homework\Homework6\Problem1.py", line 77, in driver
    [xstar,ier,its] = Newton(x0_2,to1,Nmax)
  File "C:\Users\tonys\Documents\APPM4600\Samour_APPM4600\Homework\Homework6\Problem1.py", line 128, in Newton
    Jinv = inv(J)
  File "C:\Users\tonys\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11.qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\numpy\linalg\linalg.py", line 561, in inv
    ainv = _umath_linalg.inv(a, signature=signature, extobj=extobj)
  File "C:\Users\tonys\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11.qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\numpy\linalg\linalg.py", line 112, in _raise_linalgerror_singular
    raise LinAlgError("Singular matrix")
numpy.linalg.LinAlgError: Singular matrix

```

For the initial guess (1,1):

Newton converged in the fewest iterations but took longest. Lazy Newton did not converge

For the initial guess (1,-1):

All three methods converged, and Lazy Newton took many more iterations than the other two. The Broyden method only required one more iteration than Newton's method.

For the initial guess (0,0):

An error message appeared saying singular matrix.

Let's check:  $(x_0, y_0) = (0, 0) \Rightarrow J_0 = J(0, 0) = \begin{bmatrix} 2(0) & 2(0) \\ e^0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$

$$\det(J_0) = (0)(1) - (0)(1) = 0 \Rightarrow \text{Singular matrix} \Rightarrow \text{Matrix is not invertible.}$$

Since the initial Jacobian cannot be inverted, none of the methods can be used.