HW6

Thursday, October 12, 2023 11:46 AM

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?

From HW6.py:

```
Problem 1)

i)
Iterations needed for Newton's Method: 8, duration ran: 1.019273281097412, approximated root: [-1.816264    0.83736783], error code: 0
Iterations needed for Lazy Newton's Method: 6, duration ran: 0.018027782440185547, approximated root: [ 5.3908000e+08 -1.4654546e+09], error code: 2
Iterations needed for Broyden's Method: 2, duration ran: 0.16736054420471191, approximated root: [ 41.54306 420.6087 ], error code: 2

ii)
Iterations needed for Newton's Method: 5, duration ran: 0.0787508487701416, approximated root: [ 1.0041687 -1.7296373], error code: 0
Iterations needed for Lazy Newton's Method: 28, duration ran: 0.04252982139587402, approximated root: [ 1.0041687 -1.7296373], error code: 0
Iterations needed for Broyden's Method: 38, duration ran: 0.06800532341003418, approximated root: [-9.642110e+19 -1.170142e+18], error code: 2

iii)
Iterations needed for Newton's Method: 1, duration ran: 0.012458086013793945, approximated root: [0. 0.], error code: 2
Iterations needed for Lazy Newton's Method: 1, duration ran: 0.010512351989746094, approximated root: [0. 0.], error code: 2
Iterations needed for Broyden's Method: 1, duration ran: 0.017000675201416016, approximated root: [nan nan], error code: 2

Error code key:
0: success
1: max iterations exceeded
2: next iteration will NaN or inf
```

For i), only Newton's converges to a root.

For ii), both Newton's and Lazy Newton's converge to a root. Lazy Newton's requires more iterations but takes less time.

For iii), none of the methods can converge with the initial guess.

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.

From HW6.py with initial guess (0, 1, 1)

```
Problem 2)
Iterations needed for Newton's Method: 4, duration ran: 0.8472788333892822, approximated root: [0.         0.10000004 1.        ], error code: 0
Iterations needed for Steepest Descent: 5, duration ran: 1.1964449882507324, approximated root: [4.6033776e-05 9.9970885e-02 9.9994141e-01], error code: 0
Iterations needed for Steepest Descent followed by Newton's Method: 2 + 3, duration ran: 0.33745694160461426, approximated root: [-4.6566129e-10  1.0000005
e-01  1.0000000e+00], error code: 0

Error code key:
0: success
1: max iterations exceeded
2: next iteration will NaN or inf
```

Newton's converges the fastest, but the combination method runs the fastest. The combination method allows for an excellent guess for Newton's method, making it converge very quickly.