Homework 3

Henry Dyer

March 5, 2024

* All code is in the 'HW4/4600hw4.py' file pushed to GitHub

Problem 1

Solution

(a) Iterate on this system numerically

```
[0.5    0.8660254]
Question 1A: the error message reads: 0
Question 1A: took this many seconds: 0.0003220438957214355
Question 1A: The number of iterations is: 33
```

It converges to a tolerance of 10^{-10} in 33 evaluations which is far more than Newton's method would require but since there are few matrix operations required per operation the time required is still low.

(b) Provide some motivation for the particular choice of the numerical 2×2 matrix in the equation above. The Jacobian of the initial nonlinear system is

$$\begin{bmatrix} 6x & -2y \\ 3y^2 - 3x^2 & 6xy \end{bmatrix} = \begin{bmatrix} 6 & -2 \\ 0 & 6 \end{bmatrix}$$

2hen this is evaluated at the starting point $x_0 = y_0 = 1$. The inverse of this matrix is the 2×2 matrix in the above equation evaluated at the starting point. Hence, the rationale behind this method is the same as the lazy Newton method.

$$\begin{bmatrix} 1/6 & 1/18 \\ 0 & 1/6 \end{bmatrix} \begin{bmatrix} 6 & -2 \\ 0 & -6 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(c) Iterate on (0.1) using Newton's method, using the same starting approximation $x_0 = y_0 = 1$, and check how well this converges.

```
[0.5    0.8660254]
Newton: the error message reads: 0
Newton: took this many seconds: 0.00012385845184326172
Newton: The number of iterations is: 5
```

The results obtained here are the same as found in $1A$, but Newton's method converges much in much fewer iterations and roughly three times faster (in time).

(d) Spot from your numerical result what the exact solution is, and then verify that analytically.

The two methods above answer $x = 0.5$ and $y = 0.8660254 \approx \frac{\sqrt{3}}{2}$ so we will check the equations with these points.

$$\begin{aligned} 3(1/2)^2 - (\sqrt{3}/2)^2 &= \frac{3}{4} - \frac{3}{4} = 0 \\ 3(1/2)(\sqrt{3}/2)^2 - 3(1/2)^3 - 1 &= \frac{9}{8} - \frac{1}{8} - 1 = 0 \end{aligned}$$

so the solution $(x, y) = (1/2, \sqrt{3}/2)$ is correct.

Problem 2

Solution

Use Newton and two quasi-Newton methods (Lazy Newton and Broyden) with different initial guesses:

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

```
[−1.81626407    0.8373678]
Q2 Newton: the error message reads: 0
Q2 Newton: took this many seconds: 0.00016924142837524415
Q2 Newton: The number of iterations is: 7

[nan    nan]
Q2 Lazy Newton: the error message reads: 1
Q2 Lazy Newton: took this many seconds: 0.0006696939468383789
Q2 Lazy Newton: The number of iterations is: 99

[−1.81626407    0.8373678]
Q2 Broyden: the error message reads: 0
Q2 Broyden: took this many seconds: 0.00026240348815917967
Q2 Broyden: The number of iterations is: 12
```

For the starting point $x = y = 1$, both Newton and Broyden converged to the same root while the Lazy Newton method returned an overflow error and terminated after reaching the maximum number of allowed iterations.

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

```
[1.00416874    − 1.72963729]
Q2 Newton: the error message reads: 0
Q2 Newton: took this many seconds: 0.00013120174407958983
Q2 Newton: The number of iterations is: 5

[1.00416874    − 1.72963729]
Q2 Lazy Newton: the error message reads: 0
Q2 Lazy Newton: took this many seconds: 0.00025315284729003905
Q2 Lazy Newton: The number of iterations is: 36

[1.00416874    − 1.72963729]
Q2 Broyden: the error message reads: 0
Q2 Broyden: took this many seconds: 0.00014444589614868165
Q2 Broyden: The number of iterations is: 6
```

For the starting point $x = 1, y = -1$, all three methods converged to the same root found by the Newton and Broyden methods in the previous part.

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

For the starting point $x = 0, y = 0$ none of the three methods converge as the first evaluation of the Jacobian matrix:

$$\begin{bmatrix} 2x & 2y \\ e^x & 1 \end{bmatrix}$$

at this starting point yields the singular matrix

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

since the rows are not linearly independent.

For this problem, Newton's method outperformed the two quasi-Newton methods, besides the third starting point where no method converged successfully, Newton's method converged for the other two starting points (and successfully found both roots) while doing so in less time and fewer iterations compared to both the Lazy Newton and Broyden methods. Note that the Broyden method also successfully found both roots in and depending on the starting point could potentially be better than Newton for this system.

Problem 3

Solution

Test the following three techniques for approximating the solution to the nonlinear system to within 10^{-6}.

*Note that all three of these methods are started with an initial guess of [0, 0, 0].

(a) Newton's Method

```
[0    0.1    1]
Q3 Newton: the error message reads: 0
Q3 Newton: took this many seconds: 0.00012273788452148436
Q3 Newton: The number of iterations is: 3
```

(b) Steepest Descent

```
[0    0.1    1]
Q3 Steepest Decent: the error message reads: 0
Q3 Steepest Decent: took this many seconds: 0.00146484375
Q3 Steepest Decent: number of iterations is: 3
```

(c) 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.

```
[0    0.1    1]
Q3 Hybrid: took this many seconds: 0.000715947151184082
```

We observe that all three methods converged to the correct point from the starting point $x = y = z = 0$. Newton's method was the fastest, followed by the hybrid method, and finally Steepest Decent being the slowest method of the three.

Problem 4

Solution

(a) Write $p(x)$ using the Lagrange polynomial basis for this problem.

The Lagrange polynomial of order n (for this problem $n = 4$ since we have $n + 1 = 5$ data points) is

$$p(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x) = \sum_{k=0}^{n} f(x_k)\prod_{i\neq k}\frac{x - x_i}{x_k - x_i}$$

Plugging in our data points and simplifying we get the Lagrange polynomial:

$$\begin{aligned}
L_{4,0} &= \frac{(x-2)(x-3)(x-5)(x-8)}{(0-2)(0-3)(0-5)(0-8)} = \frac{x^4 - 18x^3 + 111x^2 - 278x + 240}{240} \\
L_{4,1} &= \frac{(x-0)(x-3)(x-5)(x-8)}{(2-0)(2-3)(2-5)(2-8)} = \frac{x^4 - 16x^3 + 79x^2 - 120x}{-36} \\
L_{4,2} &= \frac{(x-0)(x-2)(x-5)(x-8)}{(3-0)(3-2)(3-5)(3-8)} = \frac{x^4 - 15x^3 + 66x^2 - 80x}{30} \\
L_{4,3} &= \frac{(x-0)(x-2)(x-3)(x-8)}{(5-0)(5-2)(5-3)(5-8)} = \frac{x^4 - 13x^3 + 46x^2 - 48x}{-90} \\
L_{4,4} &= \frac{(x-0)(x-2)(x-3)(x-5)}{(8-0)(8-2)(8-3)(8-5)} = \frac{x^4 - 10x^3 + 31x^2 - 30x}{720}
\end{aligned}$$

Plugging these into the above Lagrange polynomial, the coefficients for each term simplify to the fo allowing polynomial.

$$p(x) = x^3 - 15x^2 + 75x - 125$$

(b) Find the coefficients for Newton interpolation and write $p(x)$ as a linear combination of Newton polynomials.
The Newton polynomial interpolation is defined as

$$p(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)\ldots(x - x_{k-1})$$

$$\begin{aligned}
f[x_0] &= -125 \\
f[x_0, x_1] &= \frac{-27 - (-125)}{2 - 0} = 49 \\
f[x_1, x_2] &= \frac{-8 - (-27)}{3 - 2} = 19 \\
f[x_2, x_3] &= \frac{0 - (-8)}{5 - 3} = 4 \\
f[x_3, x_4] &= \frac{27 - 0}{8 - 5} = 9 \\
f[x_0, x_1, x_2] &= \frac{19 - 49}{3 - 0} = -10 \\
f[x_1, x_2, x_3] &= \frac{4 - 19}{5 - 2} = -5 \\
f[x_2, x_3, x_4] &= \frac{9 - 4}{8 - 3} = 1 \\
f[x_0, x_1, x_2, x_3] &= \frac{-5 - (-10)}{5 - 0} = 1 \\
f[x_1, x_2, x_3, x_4] &= \frac{-1 - (-5)}{8 - 2} = 1 \\
f[x_0, x_1, x_2, x_3, x_4] &= \frac{1 -)}{8 - 0} = 0
\end{aligned}$$

Finally, plugging divided differences back into the initial formula we obtain:

$$p(x) = -125 + 49(x) - 10(x)(x - 2) + 1(x)(x - 2)(x - 3) = (x - 5)^3$$

(c) This data was sampled from $f(x) = (x - 5)^3$. Explain what this means in terms of differences of order 4 or higher using Newton's interpolation.

Since this data was sampled from a third-order polynomial, all differences over order 4 will be zero (as seen in the above equation) so it will recover the exact polynomial with the coefficient ahead of all polynomials of a higher degree being zero.