Introduction to Numerical Algorithms

Assignment 1

Fedir Vasyliev

26/09/2024

1

1.1 A

1.1.1 A.1

Answer number 4.

1.1.2 A.2

Answer number 3.

1.2 B

Answer number 2.

1.3 C

Answer number 2.

2

All the code used for the assignment is stated in Appendix

2.1

The right-hand side has a dimension of $[3, 1]$, the system matrix is $[3, 3]$.

System matrix:

$$\begin{bmatrix} 2.127 & -6.360 & 1.135 \\ -6.360 & 354.0 & 30.00 \\ 1.135 & 30.00 & 5.000 \end{bmatrix}$$

Right-hand side matrix:

$$\begin{bmatrix} -5.225 \\ 384.0 \\ 35.00 \end{bmatrix}$$

2.2

The coefficients are $[0, 1, 1]$. We can see that the coefficient a is 0, which means that the function is purely quadratic based on the given data.

2.3

$S = 7.44 \cdot 10^{-15}$

3

3.1 Question 1

It does not converge quadratically. The left hand side represents the $|r - x_n|$ error.

```
[[2.5999999999999996, None],
 [2.5999999999999996, 0.3846153846153847],
 [0.4952380952380957, 0.07326007326007335],
 [0.2359558316080057, 0.03490470881775233],
 [0.11551269701365019, 0.47097909932525556],
 [0.05718384982464775, 1.027098179954577],
 [0.028453699146939293, 2.13245211292587 6],
 [0.014192874016099921, 4.340338160257631],
 [0.007088013884599231, 8.754817335082256],
 [0.00354190989545744, 17.583155353097872],
 [0.001770431770896419, 35.239527390130704],
 [0.0008850852264234987, 70.55212096372878],
 [0.00044250996532690934, 141.17723323025163],
 [0.00022124682279933694, 282.4274204145546]]
```

3.2 Question 2

Using the fact that $f(r) = 0$ and $f^{'}(r) = 0$ we can simplify the expression for e_{n+1}.

$$e_{n+1} = e_n - 2\frac{f(r) + f^{'}(r)e_n + \frac{1}{2}f^{''}(r)e_n^2 + \frac{1}{6}f^{'''}(\xi_n)e_n^3}{f'(r) + f''(r)e_n^2 + \frac{1}{2}f'''(\zeta_n)e_n^2}$$

$$e_{n+1} = (\frac{\frac{1}{2}f^{'''}(\zeta_n) - \frac{1}{6}f^{'''}(\xi_n)}{\frac{1}{2}f'''(\zeta_n)e_n + f''(r)})e_n^2$$

3.3 Question 3

It now converges quadratically, and the e_n/e_{n-1}^2, which is the column on the right, converges to 0.

```
[[2.5999999999999996, None],
 [2.5999999999999996, 0.3846153846153847],
 [1.6095238095238082, 0.23809523809523797],
 [0.15393860996577446, 0.022771983722747708],
 [0.0019015764605783136, 0.0007340387408660741],
 [3.0118956928504304e-07, 1.2709978213515224e-05],
 [7.549516567451064e-15, 2.087812806465376e-09]]
```

3.4 Question 4

$$f(x) = (x-2)^2(x-8)$$

$$f^{'}(x) = (2(x-2)(x-8) + (x-2)^2)$$

$$f^{''}(x) = 4x - 10 + 2(x-2) = 6x - 14$$

$$f^{'''}(x) = 6$$

As $n \to \infty$ $e_n \to 0$. Thus, the constant we found in question 2 will be

$$\frac{\frac{1}{2}\cdot 6 - \frac{1}{3}\cdot 6}{0 + 6\cdot 2 - 14} = -\frac{1}{2}$$

4 Appendix

4.1 Section 2

```
x = np.array([0, 1, 2, 3, 4]).T

y = np.array([1, 2, 5, 10, 17]).T

A = np.array([np.sin(x), x**2, np.ones(len(x))]).T

y = np.array([y]).T

c = np.linalg.solve(A.T@A,A.T@y)


f = lambda x: c[0]*np.sin(x) + c[1]*x**2 + c[2]

err = np.abs(y.T-f(x))

sm = 0

for i in range(5):

    sm += err[0][i]

sm
```

4.2 Section 3

```
def newton(f, df, x0, nmax):

    x = x0
```

```
    iterates = [x0]

    for n in range(nmax):

        x = x - f(x)/df(x)

        iterates.append(x)

    return iterates


def newton_convergence(r, X):

    error = [[abs(r-X[0]), None]]

    for i in range(len(X)):

        if i == 0: pass

        en = abs(r-X[i])

        error.append([en,abs(en/error[i-1][0]**2)])

    return error


f = lambda x: (x-2)**2*(x-8)

df = lambda x: 2*(x-2)*(x-8) + (x-2)**2

x0 = 4.6

n_max = 12
```

```
X = newton(f, df, x0, n_max)


def newton_modified(f, df, x0, nmax):

    x = x0

    iterates = [x0]

    for n in range(nmax):

        x = x - 2*f(x)/df(x)

        iterates.append(x)

    return iterates
```