Math 131A Homework 1

Jiaping Zeng

8/3/2020

Q1 (a) Continuity: Since $f(t,y) = \dfrac{1+y}{t}$ is only undefined at $t = 0$ and the given interval $t \in [1,2]$ does not include 0, $f(t,y)$ is continuous on the given interval.
Lipschitz: $\dfrac{\delta f(t,y)}{\delta y} = \dfrac{1}{t} \leq \dfrac{1}{1} = 1 = L.$
Therefore this IVP is well-posed by Theorem 5.6.

(b) Continuity: $f(t,y) = y\,cos(t)$ is continuous on $t \in [0,1]$ because both y and $cos(t)$ are defined everywhere.
Lipschitz: $\dfrac{\delta f(t,y)}{\delta y} = cos(t) \leq cos(0) = 1 = L.$
Therefore this IVP is well-posed by Theorem 5.6.

Q2 (a) $y(1.5) \approx 2 + 0.5 \cdot \dfrac{1+1}{1+2} = \dfrac{7}{3} \approx 2.333333$
$y(2) \approx \dfrac{7}{3} + 0.5 \cdot \dfrac{1+1.5}{1+\frac{7}{3}} = \boxed{\dfrac{65}{24}} \approx 2.708333$

(b)

h	y(2): Euler
0.5	2.7083333333333335
0.2	2.729166194327493
0.1	2.7355407599225927
0.01	2.741056919124695

```
def q2b():
    h_vals = [0.5, 0.2, 0.1, 0.01]  # step sizes
    t_0, t_f, y_0 = 1, 2, 2  # initial/final values

    for h in h_vals:
        t, y, steps = t_0, y_0, int((t_f-t_0)/h)
        for _ in range(steps):
            y_prev = y
            y = y_prev + h*(1+t)/(1+y_prev)
            t += h
        print(f"h={h}, y={y}")
```

(c) Exact solution: $y(2) = \sqrt{14} - 1 \approx 2.741657387$; we can see that, as expected, smaller h values resulted in approximations that are closer to the exact $y(2)$ value.

Q3 (a) $\frac{df(t,y)}{dt} = \frac{\delta f(t,y)}{\delta t} + \frac{\delta f(t,y)}{\delta y} \cdot \frac{\delta y}{\delta t} = -y^2e^{-t} + 2ye^{-t} \cdot y^2e^{-t} = \boxed{-y^2e^{-t} + 2y^3e^{-2t}}$

(b) $y(0.5) \approx 1 + 0.5 \cdot (1^2 \cdot e^0) + 0.125 \cdot (-1^2 \cdot e^0 + 2 \cdot 1^3 \cdot e^0) = 1 + 0.5 + 0.125 = 1.625$

$y(1) \approx 1.625 + 0.5 \cdot (1.625^2 \cdot e^{-0.5}) + 0.125 \cdot (-1.625^2 \cdot e^{-0.5} + 2 \cdot 1.625^2 \cdot e^{-1}) \approx \boxed{2.620252}$

(c)

h	y(1): Euler	y(1): Taylor-O2
0.5	2.1823469921767127	2.620251616284609
0.1	2.5318870490201966	2.711460067794978
0.01	2.6955194578645645	2.7182049559385204

```
def q3c():
    h_vals = [0.5, 0.1, 0.01]  # step sizes
    t_0, t_f, y_0 = 0, 1, 1  # initial/final values

    print("(Euler's method)")
    for h in h_vals:
        t, y, steps = t_0, y_0, int((t_f-t_0)/h)
        for _ in range(steps):
            y_prev = y
            y = y_prev + h*pow(y_prev, 2)*exp(-t)
            t += h
        print(f"h={h}, y={y}")

    print("(Taylor method)")
    for h in h_vals:
        t, y, steps = t_0, y_0, int((t_f-t_0)/h)
        for _ in range(steps):
            y_prev = y
            y = y_prev + h*pow(y_prev, 2)*exp(-t) + pow(h, 2) * \
                (-1*pow(y_prev, 2)*exp(-t)+2*pow(y_prev, 3)*exp(-2*t))/2
            t += h
        print(f"h={h}, y={y}")
```

(d)

h	Error: Euler	Error: Taylor-O2
0.5	0.5359348362823324	0.0980302121744363
0.1	0.18639477943884852	0.006821760664067256
0.01	0.022762370594480608	7.687252052468452e-05

As shown in the table above, the error for both methods decreases as h decreases. In addition, Taylor method of order 2 converges much faster than Euler's method as expected.