

# Math 151B Homework 3

Jiaping Zeng

8/25/2020

Q1 (a) We can first list out the first few  $w_i$  using the implicit formula as follows:

$$x_0 = 0, w_0 = 0$$

$$x_1 = h, w_1 = 0 + h(h - h^2) = h^2 - h^3$$

$$x_2 = 2h, w_2 = h^2 - h^3 + h(2h - 4h^2) = 3h^2 - 5h^3$$

$$x_3 = 3h, w_3 = 3h^2 - 5h^3 + h(3h - 9h^2) = 6h^2 - 14h^3$$

$$x_4 = 4h, w_4 = 6h^2 - 14h^3 + h(4h - 16h^2) = 10h^2 - 30h^3$$

...

Then we can see that:

$$x_i = ih, w_i = \sum_{n=1}^i (nh^2 - n^2h^3) = h^2 \left( \frac{i(i+1)}{2} - h \left( \frac{i(i+1)(2i+1)}{6} \right) \right)$$

(b)  $|w_i - y(x)| =$

$$= \left| h^2 \left( \frac{i(i+1)}{2} - h \left( \frac{i(i+1)(2i+1)}{6} \right) \right) - y(x) \right|$$

$$\leq \left| h^2 \left( \frac{i(i+1)}{2} - h \left( \frac{i(i+1)(2i+1)}{6} \right) \right) \right| + |y(x)|$$

Then, since  $h \rightarrow 0$ ,  $\left| h^2 \left( \frac{i(i+1)}{2} - h \left( \frac{i(i+1)(2i+1)}{6} \right) \right) \right| \rightarrow 0$  as well. Therefore  $|w_i - y(x)| \rightarrow |y(x)|$  as  $h \rightarrow 0$ .

Q2 In summary, under the same problem, RKF12 takes a lot more steps than RKF45 to achieve the desired tolerance. In the below code, the function used is  $f(t, y) = y - t^2 + 1$ , with  $0 \leq t \leq 2$  and tolerance  $TOL = 10^{-5}$ . The RKF45 algorithm is able to achieve this tolerance using  $0.01 \leq h \leq 0.25$  in 10 steps, whereas the RKF12 algorithm requires 174709 steps using  $0.000001 \leq h \leq 0.25$  (reduction of  $h_{min}$  was needed to achieve the same accuracy). One advantage of RKF12 is that it only needs to compute  $f$  twice per iteration versus six in RKF45, though in this case it is clear that the number of iteration it requires to achieve the same accuracy far outnumbers the evaluations it saves per iteration.

```
def rkf12():
    a, b = 0, 2 # endpoints
    y_0 = 0.5 # initial condition
    TOL = 10**-5 # tolerance
    hmin, hmax = 0.000001, 0.25 # min/max step sizes
    def f(t, y): return y-t**2+1

    print(f"TOL={TOL}")
```

```

t, w, h = a, y_0, hmax
FLAG = 1
count = 1

while FLAG:
    k1 = h*f(t, w)
    k2 = h*f(t+h, w+k1)

    R = 1/h*abs(k1-1/2*(k1+k2))

    if R <= TOL:
        t += h
        w += k1
        print(f"t={t}, w={w}, h={h}")
        count += 1

    d = 0.84*((TOL/R)**(1/4))
    if d <= 0.1:
        h /= 10
    elif d >= 4:
        h *= 4
    else:
        h *= d

    if h > hmax:
        h = hmax

    if t >= b:
        FLAG = 0
    elif t+h > b:
        h = b-t
    elif h < hmin:
        FLAG = 0
        print(f"minimum h exceeded (h={h})")

print(f"count={count}")

```