MATH 151B - Applied Numerical Methods - Homework 2

Darren Tsang, 405433124, Discussion 1A

Produced on Friday, Aug. 21 2020 @ 05:42:29 PM

Question 1

Here is Taylor's Method:

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2!}y''(t) + \frac{h^3}{3!}y^{(3)}(t) + \frac{h^4}{4!}y^{(4)}(t) + O(h^5)$$

This means that:

$$w_0 = \alpha,$$

$$w_{i+1} = w_i + hf(t_i, y(t_i)) + \frac{h^2}{2}f'(t_i, y(t_i)) + \frac{h^3}{6}f''(t_i, y(t_i)) + \frac{h^4}{24}f^{(3)}(t_i, y(t_i))$$

Applied to the IVP we are given:

$$w_0 = 0,$$

$$w_{i+1} = w_i + h(t_i^2 - 1) + \frac{h^2}{2}(2t_i) + \frac{h^3}{6}(2) + \frac{h^4}{24}(0) = w_i + h(t_i^2 - 1) + h^2 t_i + \frac{h^3}{3}$$

Part B

We begin with the following:

$$y(0) = w_0 = 0$$

Then, for the next step:

$$y(1) \approx w_1 = 0 + 1(0^2 - 1) + (1^2 * 0) + \frac{1^3}{3} = -1 + \frac{1}{3} = \frac{-2}{3}$$

The error is $|-2/3+2/3| = 0$, which means we got the exact solution. This makes sense because $f^{(3)}(t, y) = 0$.

Question 2

Part A

We are given the following:

$$a_1 f(t,y) + a_2 f(f+\alpha, y+\beta f(t,y)) \tag{1}$$

We expand (1):

$$a_1 f(t,y) + a_2[f(t, y+\beta f(t,y)) + \alpha f_t(t, y+\beta f(t,y)) + O(\alpha^2)]$$

Distributing a_2:

$$a_1 f(t,y) + a_2 f(t, y+\beta f(t,y)) + a_2\alpha f_t(t, y+\beta f(t,y)) + O(\alpha^2) \tag{2}$$

Let $I = a_2 f(t, y+\beta f(t,y))$. Then:

$$I = a_2[f(t,y) + \beta f(t,y) f_y(t,y) + O(\beta^2)] = a_2 f(t,y) + a_2\beta f(t,y) f_y(t,y) + O(\beta^2)$$

Let $J = a_2\alpha f_t(t, y+\beta f(t,y))$. Then:

$$J = a_2\alpha[f_t(t,y) + \beta f(t,y) f_{ty}(t,y) + O(\beta^2)] = a_2\alpha f_t(t,y) + a_2\alpha\beta f(t,y) f_{ty}(t,y) + O(\beta^2)$$

Plugging in I and J into (2) (and setting $a_2\alpha\beta f(t,y) f_{ty}(t,y) = O(\alpha\beta)$):

$$a_1 f(t,y) + a_2 f(t,y) + a_2\beta f(t,y) f_y(t,y) + a_2\alpha f_t(t,y) + O(\alpha\beta) + O(\beta^2) + O(\alpha^2)$$

We can rewrite as:

$$(a_1 + a_2) f(t,y) + a_2\beta f(t,y) f_y(t,y) + a_2\alpha f_t(t,y)$$

Then, to match the equation given to us:

$$a_1 + a_2 = 1, a_2\beta = a_2\alpha = \frac{h}{2}$$

A possible solution is:

$$a_1 = a_2 = \frac{1}{2}, \alpha = \beta = h$$

Part B

If we set $a_1 = 1/2$, then $a_1 = a_2 = 1/2$ and $\alpha = \beta = h$. We get:

$$T^{(2)}(t,y) \approx \frac{1}{2} f(t,y) + \frac{1}{2} f(t+h, y+hf(t,y))$$

Then, we get:

$$w_i = w_{i-1} + hT^{(2)}(t_{i-1}, w_{i-1}) = w_{i-1} + h\left[\frac{1}{2} f(t_{i-1}, w_{i-1}) + \frac{1}{2} f(t_i, w_{i-1} + hf(t_{i-1}, w_{i-1}))\right],$$

which is Modified Euler's Method, as desired.

Part C

We begin with the following:

$$y(t+h) = y(t) + h\left[\frac{1}{2}f(t,y) + \frac{1}{2}f(t+h, y+hf(t,y)) + O(\alpha\beta) + O(\beta^2) + O(\alpha^2)\right]$$

Since $\alpha = \beta = h$:

$$y(t+h) = y(t) + h\left[\frac{1}{2}f(t,y) + \frac{1}{2}f(t+h, y+hf(t,y)) + O(h^2)\right]$$

Rearranging:

$$y(t+h) = y(t) + h\left[\frac{1}{2}f(t,y) + \frac{1}{2}f(t+h, y+hf(t,y))\right] + O(h^3)$$

Getting into the right form:

$$O(h^2) = \frac{y(t+h) - y(t)}{h} - \left[\frac{1}{2}f(t,y) + \frac{1}{2}f(t+h, y+hf(t,y))\right]$$

We can see the truncation error above. Also, we can see that the error is of order 2.

Part D

We start with the initial condition:

$$y(0) = w_0 = 0$$

For the next step,

$$y(.5) \approx w_1 = 0 + \frac{.5}{2}[(0^2 - 1) + (.5^2 - 1)] = -.4375$$

For the next step,

$$y(1) \approx w_2 = -.4375 + \frac{.5}{2}[(.5^2 - 1) + (1^2 - 1)] = -.625$$

Question 3

Part A

We are given the following:

$$w_{i+1} = w_i + \frac{h}{4}\left[f(t_i, w_i) + 3\left\{f\left(t_i + \frac{2h}{3}, w_i + \frac{2h}{3} f\left(t_i + \frac{h}{3}, w_i + \frac{h}{3} f(t_i, w_i)\right)\right)\right\}\right]$$

We will rewrite as:

$$w_{i+1} = w_i + \frac{h}{4}\left[f(t_i, w_i) + 3\alpha\right],$$

where $\alpha = f(t_i + \frac{2h}{3}, w_i + \frac{2h}{3}\beta)$ and $\beta = f(t_i + \frac{h}{3}, w_i + \frac{h}{3} f(t_i, w_i))$

```
def heuns_method(func, start, end, initial, h):
  N = (end - start)/h
  t = [start]
  w = [initial]
  for i in range(1, round(N+1)):
    beta = func(t[i-1] + h/3, w[i-1] + h*func(t[i-1], w[i-1])/3)
    alpha = func(t[i-1] + 2*h/3, w[i-1] + 2*h*beta/3)
    w.append(w[i-1] + h*(func(t[i-1], w[i-1]) + 3*alpha)/4)
    t.append(start + i*h)

  return(w[-1])
```

Part B

```
def euler(func, start, end, initial, h):
  N = (end - start)/h
  t = [start]
  w = [initial]

  for i in range(1, round(N+1)):
    w.append(w[i-1] + h*func(t[i-1], w[i-1]))
    t.append(start + i*h)

  return(w[-1])
```

```
def modified_euler(func, start, end, initial, h):
  N = (end - start)/h
  t = [start]
  w = [initial]
  for i in range(1, round(N+1)):
    a = func(t[i-1], w[i-1])
    b = func(t[i-1] + h, w[i-1] + h*func(t[i-1], w[i-1]))
    w.append(w[i-1] + h/2*(a + b))
    t.append(start + i*h)

  return(w[-1])
```

```python
import matplotlib.pyplot as plt

def plotter(euler, modified, huen, h, actual):
  euler = [abs(x - actual) for x in euler]
  modified = [abs(x - actual) for x in modified]
  huen = [abs(x - actual) for x in huen]

  plt.semilogy(h, euler, label="Euler's Method")
  plt.semilogy(h, modified, label="Modified Euler's Method")
  plt.semilogy(h, huen, label="Huen's Method")
  plt.ylabel('log(Absolute Error)')
  plt.xlabel('Value of h')
  plt.title('Comparison of Errors for Different Methods with Varying Step Sizes')
  plt.legend()
  plt.show()
```

```python
def driver(func, start, end, initial, all_h, actual):
  euler_approx = []
  modified_euler_approx = []
  huen_approx = []

  for h in all_h:
    x = euler(func, start, end, initial, h)
    y = modified_euler(func, start, end, initial, h)
    z = heuns_method(func, start, end, initial, h)

    euler_approx.append(x)
    modified_euler_approx.append(y)
    huen_approx.append(z)

  plotter(euler_approx, modified_euler_approx, huen_approx, all_h, actual)
```

We will use the following IVP:

$$\frac{dy}{dt} = t^2 - y - 1, \qquad 0 \leq t \leq 1, \qquad y(0) = 0$$

The actual solution is:

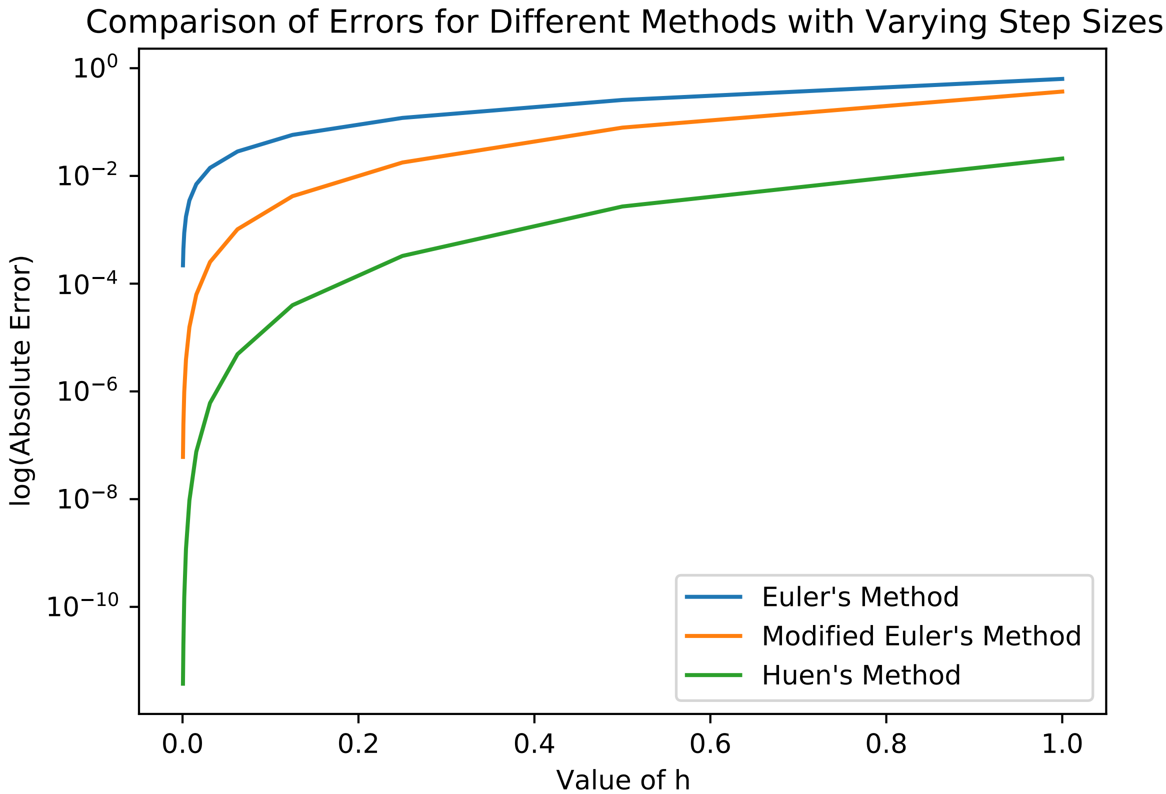
$$y(t) = t^2 - 2t + 1 - e^{-t}$$

Thus, at $t = 1$:

$$y(1) = (1)^2 - 2(1) + 1 - e^{-1} = -e^{-1} \approx -.36787944117$$

```python
def func(t, y):
  return(t*t - y - 1)

all_h = [1/(2**x) for x in range(11, -1, -1)]
driver(func, 0, 1, 0, all_h, -.36787944117)
```



As we can see from the graph above, the error increases as h increases for all methods. Furthermore, for the same h values, the error for Huen's Method $<$ the error for Modified Euler's Method $<$ the error for Euler's Method, as expected.

Question 17 (from textbook)

We are given the following:

$$\frac{dx(t)}{dt} = (b-d)x(t) \longrightarrow \frac{x'(t)}{x(t)} = (b-d), \tag{1}$$

$$\frac{dx_n(t)}{dt} = (b-d)x_n(t) + rb(x(t) - x_n(t)) \tag{2}$$

Part A

We can rewrite $p(x) = x_n(t)/x(t)$:

$$x_n(t) = p(t)x(t) \longrightarrow x_n'(t) = p'(t)x(t) + p(t)x'(t)$$

Plugging in $x_n(t)$ and $x_n'(t)$ into (2):

$$p'(t)x(t) + p(t)x'(t) = (b-d)p(t)x(t) + rb(x(t) - p(t)x(t))$$

Rearranging and dividing both sides by $x(t)$:

$$p'(t) = (b-d)p(t) + rb(1-p(t)) - \frac{p(t)x'(t)}{x(t)}$$

From (1):

$$p'(t) = (b-d)p(t) + rb(1-p(t)) - p(t)(b-d)$$

Canceling some terms:

$$p'(t) = \frac{dp}{dt} = rb(1-p(t)),$$

as desired.

Part B

```
def func(t, p):
  return(.1*.02*(1-p))
```

```
print(euler(func, 0, 50, .01, h = 1))
```

```
## 0.10430065017600461
```

```
print(modified_euler(func, 0, 50, .01, h = 1))
```

```
## 0.10421089633547789
```

```
print(heuns_method(func, 0, 50, .01, h = 1))
```

```
## 0.1042109561743075
```

We try all 3 methods, and it seems that $p(50) \approx .104$.

Part C

Using a calculator to solve for $p(t)$:

$$p(t) = 1 - .99e^{-rbt}$$

Plugging in $t = 50$ into $p(t)$:

$$p(50) = 1 - .99e^{-50(.1)(.02)} = .1042109561$$

The absolute error is $|.10430065017600461 - .1042109561| = 8.969408 \cdot 10^{-05}$, $|.10421095554711789 - .1042109561| = -5.528821 \cdot 10^{-10}$, and $|.10421095614442985 - .1042109561| = 4.442985 \cdot 10^{-11}$ for Euler's, Modified Euler's, and Huen's Method respectively.

As expected, the error for Huen's Method is the lowest, but all of them are pretty low.