

3940 AS05

Main

Question 1

Which of the following statements are **TRUE**?

- (i) The differential equation $2y' + x^2y = 2x + 3$, $y(0) = 5$ is linear.
- (ii) A differential equation is considered to be ordinary if it has one dependent variable.
- (iii) The exact of ordinary equation, $2y' + 3y = e^{-x}$, $y(0) = 1$ is $y(x) = e^{-x}$.
- (iv) The initial value problem, $y' = y - t \cos ty$, $0 \leq t \leq 2$, $y(0) = 2$ has a unique solution.

Solution 1

(ii), (iii)

Question 2

Given $3y' + 5y^2 = \sin x$, $y(0.3) = 5$ and using a step size of $h = 0.3$, the value of $y(0.9)$ using Euler's method (correct to three decimal places)?

Solution 2

For this problem $f(t, y) = \frac{1}{3}(\sin x - 5y^2)$, so

$$w_0 = y(0.3) = 5$$

$$w_1 = w_0 + 0.1(\sin 0.3 - 5w_0^2) = 5 + 0.1(\sin 0.3 - 125) = -7.4704479793$$

$$y(0.9) = w_2 = w_1 + 0.1(\sin 0.6 - 5w_1^2) = -35.3177802377$$

Question 3

Euler's method can be derived by using the first two terms of the Taylor series of writing the value of y_{i+1} , that is the value of y at x_{i+1} , in terms of y_i and all the derivatives of y at x_i . If $h = x_{i+1} - x_i$, then write down the explicit expression for y_{i+1} if the first three terms of the Taylor series are chosen for the ordinary differential equation, $3y' + 2y = e^{-4x}$, $y(0) = 7$

Solution 3

Given

$$3y' + 2y = e^{-4x}, y(0) = 7$$

$$\text{So, } \frac{dy}{dx} = \frac{1}{3}(e^{-4x} - 2y)$$

Then, we have

$$\frac{d^2y}{dx^2} = \frac{1}{3}(-4e^{-4x} - 2\frac{dy}{dx}) = -\frac{4}{3}e^{-4x} - \frac{2}{3} \cdot \frac{1}{3}(e^{-4x} - 2y) = -\frac{14}{9}e^{-4x} + \frac{4}{9}y$$

By Taylor series,

$$\begin{aligned}
y_{i+1} &= y_i + h \frac{dy}{dx}|_i + \frac{h^2}{2} \frac{d^2y}{dx^2}|_i + O(h^3) \\
&= y_i + h \cdot \frac{1}{3} (e^{-4x} - 2y) + \frac{h^2}{2} \cdot \left(-\frac{14}{9e^{-14x}} + \frac{4}{9}y \right) + O(h^3)
\end{aligned}$$

Question 4

Given the initial-value problem $\frac{dy}{dt} = \frac{1}{t^2} - \frac{y}{t} - y^2$, $1 \leq t \leq 2$, $y(1) = -1$, with exact solution $y(t) = -1/t$. Use Taylor's method of order two with $h = 0.05$ to approximate the solution and compare it with the actual value at $y(1.1)$.

Solution 4

For the method of order 2, we need the first derivative of $f(t, y(t)) = \frac{1}{t^2} - \frac{y(t)}{t} - y^2$ with respect to the variable t . Because $\frac{dy}{dt} = \frac{1}{t^2} - \frac{y}{t} - y^2$, we have

$$\begin{aligned}
f'(t, y(t)) &= -\frac{2}{t^3} + \frac{y'}{t^2} - 2yy' \\
&= -\frac{2}{t^3} + \frac{\frac{1}{t^2} - \frac{y}{t} - y^2}{t^2} - 2y\left(\frac{1}{t^2} - \frac{y}{t} - y^2\right) \\
&= \frac{1}{t^4} - \frac{2-y}{t^3} - \frac{y^2+2y}{t^2} + \frac{2y^2}{t} + 2y^3
\end{aligned}$$

so

$$\begin{aligned}
T^{(2)}(t_i, w_i) &= f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) \\
&= \frac{1}{t^2} - \frac{w_i}{t} - w_i^2 + \frac{h}{2} \left(\frac{1}{t^4} - \frac{2-w_i}{t^3} - \frac{w_i^2+2w_i}{t^2} + \frac{2w_i^2}{t} + 2w_i^3 \right)
\end{aligned}$$

Given $h = 0.05$, and $t_i = 1 + 0.05i$ for each $i = 1, 2, 3, \dots$. Thus, the second-order method becomes

$$\begin{aligned}
w_0 &= -1 \\
w_{i+1} &= w_i + h \times T^{(2)}(t_i, w_i) = w_i + 0.05 \times T^{(2)}(1 + 0.05i, w_i)
\end{aligned}$$

Therefore, the approximations are (calculated by the code below)

$$\begin{aligned}
w_1 &= -0.95125 \\
y(1.1) \approx w_2 &= -0.906869457
\end{aligned}$$

The actual value of $y(1.1)$ is -0.9090909090 , so the difference between them is 0.0022214520

```

1 import Foundation
2
3 let h = 0.05
4 let t: (Double, Double) -> Double = { t, w in
5     let f = 1/pow(t, 2) - w/t - pow(w, 2)
6     let f_d = 1/pow(t, 4) - (2-w)/pow(t, 3) - (w*(w+2))/pow(t, 2) + 2*pow(w, 2)/t + 2*pow(w, 3)
7
8     return f + h/2*(f_d)
9 }
10
11 var w: [Double] = [-1.0]
12 for i in 0...1 {
13     w.append(w[i]+h*t(1+0.05*Double(i), w[i]))
14 }
15 print(w)
16 print(-1/1.1)

```

(Implemented with the Swift Programming Language, use [this online Swift compiler](#) to execute the code if needed --automatically generated by Wendell's Custom Markdown Compiler)

Question 5

Given $3y' + 5\sqrt{y} = e^{0.1x}$, $y(0.3) = 5$ and using a step size of $h = 0.3$, the best estimate of $y'(0.6)$ using the Runge-Kutta 2nd order midpoint method (Correct to two decimal places)?

Solution 5

From the question, we could know that

$$t_i = 0.3 + 0.3i, y' = \frac{1}{3}(e^{0.1x} - 5\sqrt{y})$$

The Runge-Kutta second order midpoint method is

$$w_o = \alpha, w_{i+1} = w_i + hf(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i))$$

Now given that,

$$\begin{aligned} w_0 &= 5 \\ y(0.6) &\approx w_1 = w_0 + hf(t_0 + \frac{h}{2}, w_i + \frac{h}{2}f(t_0, w_0)) \\ &= 5 + 0.3 \times f(0.45, 5 + 0.15 \times f(0.3, 5)) \\ &= 4.044826 \end{aligned}$$

Therefore,

$$y'(0.6) = f(0.6, y(0.6)) = \frac{1}{3}(e^{0.1 \times 0.6} - 5y(0.6)) \approx -2.998013 \approx -3.00$$

```
1 import Foundation
2
3 let f: (Double, Double) -> Double = { t, w in
4     1/3*(pow(exp(1), 0.1*t) - 5*(pow(w, 0.5)))
5 }
6
7 let y = 5 + 0.3 * f(0.45, 5 + 0.15 * f(0.3, 5))
8
9 print(f(0.6, y))
```

(Implemented with the Swift Programming Language, use [this online Swift compiler](#) to execute the code if needed --automatically generated by Wendell's Custom Markdown Compiler)

Question 6

Use the Modified Euler method to approximate the solutions to following initial-value problems and compare the results to the actual value:

$$\frac{dy}{dt} = \frac{y^2}{1+t}, 1 \leq t \leq 2, y(1) = \frac{-1}{\ln 2} \text{ with } h = 0.1; \text{ actual solution } y(t) = \frac{-1}{\ln(t+1)}.$$

Solution 6

For the initial-value problem $y' = \frac{y^2}{1+t}$, the given solution is $y(t) = -\frac{1}{\ln(t+1)}$.

Compare the given form with the general form of the initial value problem,

$$y' = f(t, y), a < t < b, y(a) = \alpha$$

to get

$$f(t, y) = \frac{y^2}{1+t}, \alpha = w_0 = y_0 = -(\ln 2)^{-1}, t_0 = a = 1, b = 2$$

Here, $t_i = t_0 + h_i = 1 + 0.1i$

The modified Euler method of order two is

$$w_0 = \alpha, w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + h \times f(t_i, w_i))]$$

Now given that,

$$w_0 = -(\ln 2)^{-1}$$

$$\text{So, } w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + h \times f(t_i, w_i))]$$

$$w_1 = w_0 + \frac{1}{20}[f(1 + 0.1 \times 0, w_0) + f(1 + 0.1 \times 1, w_0 + h \times f(1 + 0.1 \times 0, w_0))]$$

$$= w_0 + \frac{1}{20}\left[\frac{1}{2 \ln^2 2} + f(1.1, w_0 + 0.1 \times f(1, w_0))\right]$$

$$w_2 = w_1 + \frac{h}{2}[f(t_1, w_1) + f(t_{1+1}, w_1 + h \times f(t_1, w_1))]$$

...

calculated with code below

```

1  import Foundation
2
3  let h = 0.1
4  let t: (Int) -> Double = { 1 + 0.1 * Double($0) }
5  let f: (Double, Double) -> Double = { pow($1, 2) / (1 + $0) }
6
7  var w: [Double] = [-1 / log(2.0)]
8  // calculate from w_1 to w_10
9  for i in 0...9 {
10     w.append(w[i] + h/2*(f(t(i), w[i]) + f(t(1), w[i]+h*f(t(i), w[i]))))
11 }
12 // print the value of w
13 for i in w.indices {
14     print("\(i): \(w[i])")
15 }
16 // the function of actual solution
17 let y: (Int) -> Double = { -1 / log(Double($0) + 1) }
18 // print the value of actual solution
19 for i in 1...2 {
20     print("\(i): \(y(i))")
21 }

```

(Implemented with the Swift Programming Language, use [this online Swift compiler](#) to execute the code if needed --automatically generated by Wendell's Custom Markdown Compiler)

$$\begin{aligned}
w_0 &= -1.4426950408889634 \\
w_1 &= -1.3479960270214981 \\
w_2 &= -1.2668437832914985 \\
w_3 &= -1.1964312454675494 \\
w_4 &= -1.1346843188166007 \\
w_5 &= -1.0800363811617362 \\
w_6 &= -1.0312813686225435 \\
w_7 &= -0.9874752117356199 \\
w_8 &= -0.9478680004287579 \\
w_9 &= -0.9118562387718296 \\
w_{10} &= -0.8789485652546548
\end{aligned}$$

The actual value of $y(2)$ is -0.9102392266268373

So, the difference between the estimated value and the actual value is $w_{10} - y(2) = -0.0312906614$

Question 7

Show that the Midpoint method and the Modified Euler method give the same approximation to the initial-value problem, $y' = -y + t + 1$, $y(0) = 1$ for any choice of h . Why is this true? What is the relationship between these methods?

Solution 7

For both method, we have $w_0 = \alpha = 1$

So, in the Midpoint method, we have

$$\begin{aligned}
w_{i+1} &= w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right) \\
&= w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}(-w_i + t_i + 1)\right) \\
&= w_i + h\left[-w_i - \frac{h}{2}(-w_i + t_i + 1) + t_i + \frac{h}{2} + 1\right] \\
&= w_i + h\left[-w_i - \frac{h}{2}(-w_i + t_i) + t_i + 1\right] \\
&= w_i + -\frac{h}{2}[-2w_i + hw_i - ht_i + 2t_i + 2]
\end{aligned}$$

In the Modified Euler Method, we have

$$\begin{aligned}
w_{i+1} &= w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))] \\
&= w_i + \frac{h}{2}[-w_i + t_i + 1 + f(t_i + h, w_i + h(-w_i + t_i + 1))] \\
&= w_i + \frac{h}{2}[-w_i + t_i + 1 - (w_i + h(-w_i + t_i + 1)) + t_i + h + 1] \\
&= w_i + \frac{h}{2}[-2w_i + hw_i - ht_i + 2t_i + 2]
\end{aligned}$$

Hence, the Midpoint method and the Modified Euler method give the same approximation to the initial-value problem

Question 8

In numerical methods, as h is decreased the calculation takes longer but is more accurate. However, decreasing h too much could cause significant error. Why does this occur?

Solution 8

As the step size decreases and the number of steps increases, arithmetic error also increases since the arithmetic error of each steps is stacked together and gradually scaled up with each iteration.

Question 9

Use the Runge-Kutta Fehlberg Algorithm with tolerance $TOL = 10^{-4}$ to approximate the solution to the following initial-value problems, $y' = \sin t + e^{-t}$, $0 \leq t \leq 1$, $y(0) = 0$ with $h_{\max} = 0.25$ and $h_{\min} = 0.02$.

Solution 9

$$t_0 = 0, w_0 = 0$$

let $h = h_{\max} = 0.25$, let k_1, \dots, k_6 be six evaluations of

$$\begin{aligned}k_1 &= hf(t_0, w_0) \\&= hf(0, 0) = 0.25 \\k_2 &= hf(t_0 + \frac{1}{4}h, w_0 + \frac{1}{4}k_1) \\&= hf(\frac{1}{4}h, \frac{1}{4}k_1) = 0.250468095 \\k_3 &= hf(t_0 + \frac{3}{8}h, w_0 + \frac{3}{32}k_1 + \frac{9}{32}k_2) \\&= hf(\frac{3}{8}h, \frac{3}{32}k_1 + \frac{9}{32}k_2) = 0.251030773 \\k_4 &= hf(t_0 + \frac{12}{13}h, w_0 + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3) \\&= hf(\frac{12}{13}h, \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3) = 0.25566271 \\k_5 &= hf(t_0 + h, w_0 + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4) \\&= hf(h, \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4) = 0.2565511856 \\k_6 &= hf(t_0 + \frac{1}{2}h, w_0 - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5) \\&= hf(\frac{1}{2}h, \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5) = 0.2571929080\end{aligned}$$

So,

$$\begin{aligned}w_{i+1} &= w_i + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 = w_i + 0.252286795 = 0.252286795i \\w_4 &= 1.0091471786\end{aligned}$$

This also implies that

$$R = 7.898628976 \times 10^{-8} \leq TOL$$

So we can accept the approximation 1.0091471786

Question 10

Use all the Adams-Bashforth methods to approximate the solutions to the initial-value problems:

$y' = 1 + y/t$, $1 \leq t \leq 2$, $y(1) = 2$ with $h = 0.2$; actual solution, $y(t) = t \ln t + 2t$. Use exact starting values and compare the results to the actual values.

Solution 10

$$w_0 = \alpha_0 = 2, w_1 = \alpha_1 = 2.6187$$

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

$$w_0 = 2.0$$

$$w_1 = 2.6187858681527456$$

$$w_2 = 3.2734823351909323$$

$$w_3 = 3.9567106799095937$$

$$w_4 = 4.66477376559329$$

$$y(2) \approx w_5 = 5.394941642364489$$

The actual value of $y(2)$ is 5.386294361119891

Hence, the difference of the actual value and approximation is
 $|5.386294361119891 - 5.394941642364489| = 0.0086472812$

```

1  import Foundation
2
3  let h = 0.2
4  let t: (Int) -> Double = { 1 + Double($0) * h }
5  let f: (Double, Double) -> Double = { t, w in
6      1 + w/t
7  }
8  let y: (Double) -> Double = { $0 * log($0) + 2 * $0 }
9  var w: [Double] = [2.0]
10 w.append(y(t(1)))
11
12 for i in 1...4 {
13     w.append(w[i] + h / 2 * ( 3 * f(t(i), w[i]) - f(t(i-1), w[i-1]) ))
14 }
15
16 print(w)
17 print(y(2))

```

(Implemented with the Swift Programming Language, use [this online Swift compiler](#) to execute the code if needed --automatically generated by Wendell's Custom Markdown Compiler)