

Problem 1.

Find the characteristic equation and reduced characteristic equation of the 3-step method

$$y_{n+1} = -\frac{27}{11}y_n + \frac{27}{11}y_{n-1} + y_{n-2} + h \left[\frac{3}{11}f_{n+1} + \frac{27}{11}f_n + \frac{27}{11}f_{n-1} + \frac{2}{11}f_{n-2} \right]$$

Find the (exact) roots of the reduced characteristic equation, and determine if this method is stable for small $h\lambda$ or not.

Solution

TODO

Problem 2.

The IVP

$$\begin{aligned}y' &= -y + 2e^{-t} \cos(2t) \\ y(0) &= 0\end{aligned}$$

has the true solution $y(t) = e^{-t} \sin(2t)$.

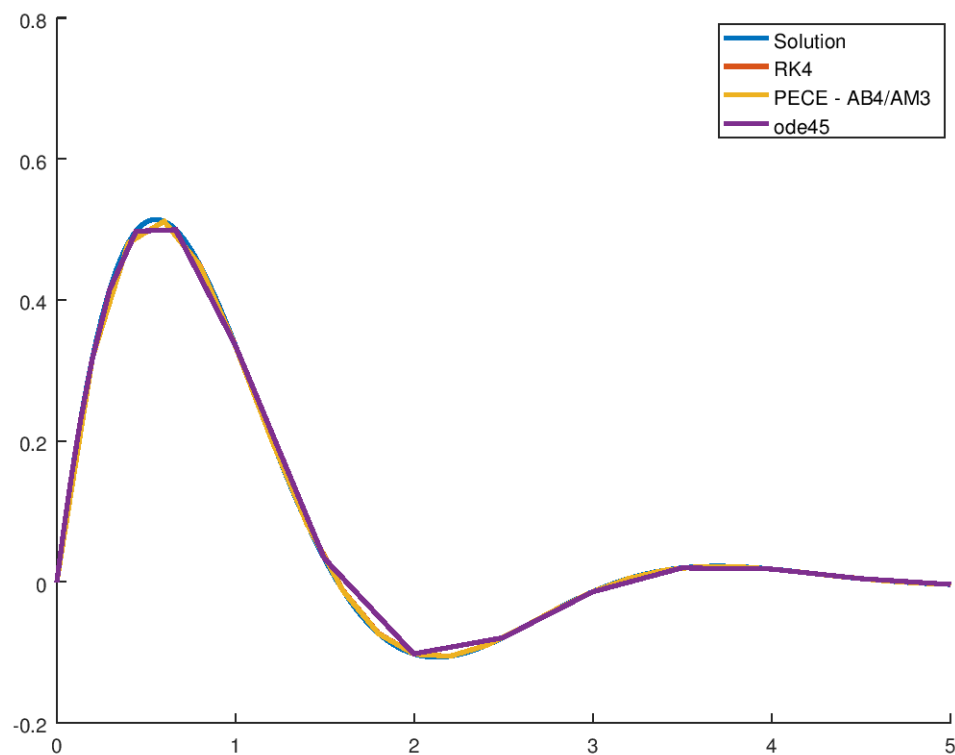
Solve it numerically from $t=0$ to $t=5$, using

- (a) the classical 4-stage Runge-Kutta method with $h=0.2$
- (b) a PECE predictor-corrector method based on 4-step AB, 3-step AM, $h=0.2$
Use RK values from part (a) for the startup.
- (c) the Matlab built-in ODE45. This routine will pick its own steps.

Plot all 3 curves in one picture.

Print out the values at $t=3$ for all methods, and the error at that point. You don't need to print out the rest of the values.

Run the RK and ABM programs again with $h=0.1$, and compare the error at $t=3$ for $h=0.2$ and $h=0.1$.

Solution

	Actual	RK4	PECE - AB4/AM3	ODE45 (h unknown)
h = 0.2				
Value	-0.013911	-0.013911	-0.013776	-0.014068
Error		$2.1822 * 10^{-7}$	$1.3560 * 10^{-4}$	$1.5663 * 10^{-4}$
h = 0.1				
Value	-0.013911	-0.013911	-0.013908	
Error		$1.2907 * 10^{-8}$	$3.3871 * 10^{-6}$	

```

1 function p2
2
3 clear all;
4 close all;
5 hold on;
6
7 % Actual solution.
8 _f = @(t) e.^(-t) .* sin(2 .* t);
9 _t = linspace(0, 5, 200);
10 _y = _f(_t);
11
12 plot(_t, _y, 'DisplayName', 'Solution', 'LineWidth', 2);
13 val = _f(3)
14
15 % Numerical Solutions
16 f = @(t, y) -y + 2 .* e.^(-t) * cos(2*t);
17 h = 0.2; % Step Size
18 t = 0:h:5; % Interval
19 y = [0]; % Initial point
20 steps = length(t);
21
22 % RK4
23 for n = 1:(steps-1)
24
25     % 4-Step
26     k1 = f(t(n), y(n));
27     k2 = f(t(n) + h/2, y(n) + (h/2) * k1);
28     k3 = f(t(n) + h/2, y(n) + (h/2) * k2);
29     k4 = f(t(n) + h, y(n) + h * k3);
30
31     % Compute y_{n+1}
32     y(n+1) = y(n) + (h/6) * (k1 + 2*k2 + 2*k3 + k4);
33
34 end
35
36 val = y((3/h) + 1)
37 err = norm(_f(3) - val, inf)

```

```
38 plot(t, y, 'DisplayName', 'RK4', 'LineWidth', 2);
39
40 % PECE
41 y = y(1:4); % Pull startup values from RK4
42
43 for n = 5:steps
44
45     % P: 4-Step AB
46     y(n) = y(n-1) + h * (
47         (55/24) * f(t(n-1), y(n-1)) -
48         (59/24) * f(t(n-2), y(n-2)) +
49         (37/24) * f(t(n-3), y(n-3)) -
50         (3/8) * f(t(n-4), y(n-4))
51     );
52
53     % C: 3-Step AM
54     y(n) = y(n-1) + h * (
55         (3/8) * f(t(n), y(n)) +
56         (19/24) * f(t(n-1), y(n-1)) -
57         (5/24) * f(t(n-2), y(n-2)) +
58         (1/24) * f(t(n-3), y(n-3))
59     );
60
61 end
62
63 val = y((3/h) + 1)
64 err = norm(_f(3) - val, inf)
65 plot(t, y, 'DisplayName', 'PECE - AB4/AM3', 'LineWidth', 2);
66
67 % ODE45
68 % Note I am using octave, so the implementation may differ slightly
69 % from the one provided by Matlab.
70
71 [solx, soly] = ode45(f, [0 5], [0 0]);
72 val = interp1(solx, soly(:,1), 3, method='linear')
73 err = norm(_f(3) - val, inf)
74 plot(solx, soly(:,1), 'DisplayName', 'ode45', 'LineWidth', 2);
75
76 legend('show');
77
78 end
```

Problem 3.

Use the Matlab routine *fzero* or something comparable to find the two solutions of

$$e^x - x - 2 = 0.$$

There is one positive and one negative solution.

Solution

TODO