

Лабораторная Работа № 1

Численные методы решения задачи Коши для ОДУ

Царик Виталий
3-й курс 2-я группа

11 октября 2019 г.

1 Постановка задачи

Условие

Задачу Коши для данного дифференциального уравнения 2-го порядка преобразовать к задаче Коши для системы двух дифференциальных уравнений 1-го порядка. Найти решение последней задачи на отрезке $[a, b]$ с шагом $\tau = 0.05$ указанными методами. Оценить погрешность численного решения с помощью правила Рунге (для одношаговых методов). Сравнить численное решение с известным аналитическим решением $u(t)$.

Задача

$$\begin{cases} t(t+1)u''(t) + (3t+2)u'(t) + u(t) = 0 \\ u(1) = 1 \\ u'(1) = -1 \\ 1 \leq t \leq 2 \end{cases} \quad (1)$$

$$u(t) = \frac{1}{t} \quad (2)$$

Методы

- 1) Явный метод трапеций
- 2) Экстраполяционный метод Адамса 2-го порядка
- 3) Неявный метод трапеций

2 Краткие теоретические сведения

Приведём исходное ДУ 2-го порядка к системе из двух ДУ, сделав замену

$$\begin{cases} u_1(t) = u(t) \\ u_2(t) = u'(t) \end{cases}$$

Получим

$$\begin{cases} u_1'(t) = u_2(t) \\ u_2'(t) = -\frac{(3t+2)u_2(t) + u_1(t)}{t(t+1)} \\ u_1(1) = 1 \\ u_2(1) = -1 \end{cases} \quad (3)$$

Явный метод трапеций

$$\begin{cases} y_1(t_{j+1}) = y_1(t_j) + 0.025(y_2(t_j) + y_2^*(t_{j+1})) \\ y_2(t_{j+1}) = y_2(t_j) - 0.025 \left(\frac{(3t_j+2)y_2(t_j) + y_1(t_j)}{t_j(t_j+1)} + \frac{(3t_{j+1}+2)y_2^*(t_{j+1}) + y_1^*(t_{j+1})}{t_{j+1}(t_{j+1}+1)} \right) \\ y_1^*(t_{j+1}) = y_1(t_j) + 0.05y_2(t_j) \\ y_2^*(t_{j+1}) = y_2(t_j) - 0.05 \frac{(3t_j+2)y_2(t_j) + y_1(t_j)}{t_j(t_j+1)} \\ y_1(1) = 1 \\ y_2(1) = -1 \end{cases}$$

Порядок точности: $o(\tau^2)$

Экстраполяционный метод Адамса 2-го порядка

$$\begin{cases} y_1(t_{j+1}) = y_1(t_j) + 0.025(3y_2(t_j) - y_2(t_{j-1})) \\ y_2(t_{j+1}) = y_2(t_j) - 0.025 \left(3 \frac{(3t_j+2)y_2(t_j) + y_1(t_j)}{t_j(t_j+1)} - \frac{(3t_{j-1}+2)y_2(t_{j-1}) + y_1(t_{j-1})}{t_{j-1}(t_{j-1}+1)} \right) \\ y_1(1) = 1 \\ y_2(1) = -1 \end{cases}$$

Порядок точности: $o(\tau^2)$

Начало таблицы будем искать с помощью явного метода трапеций

Неявный метод трапеций

$$\begin{cases} y_1(t_{j+1}) = y_1(t_j) + 0.025(y_2(t_j) + y_2(t_{j+1})) \\ y_2(t_{j+1}) = y_2(t_j) - 0.025 \left(\frac{(3t_j + 2)y_2(t_j) + y_1(t_j)}{t_j(t_j + 1)} + \frac{(3t_{j+1} + 2)y_2(t_{j+1}) + y_1(t_{j+1})}{t_{j+1}(t_{j+1} + 1)} \right) \\ y_1(1) = 1 \\ y_2(1) = -1 \end{cases}$$

Порядок точности: $o(\tau^2)$

В данном случае, система получилась линейной относительно $y_1(t_{j+1})$, $y_2(t_{j+1})$. Поэтому можем решать её методом Гаусса

$$Ay = b$$

где,

$$y = (y_1(t_{j+1}), y_2(t_{j+1}))^T$$

$$\begin{cases} a_{1,1} = 1 \\ a_{1,2} = -0.025 \\ a_{2,1} = \frac{0.025}{t_{j+1}(t_{j+1} + 1)} \\ a_{2,2} = 1 + 0.025 \left(\frac{3t_{j+1} + 2}{t_{j+1}(t_{j+1} + 1)} \right) \\ b_1 = y_1(t_j) + 0.025y_2(t_j) \\ b_2 = y_2(t_j) - 0.025 \left(\frac{(3t_j + 2)y_2(t_j) + y_1(t_j)}{t_j(t_j + 1)} \right) \end{cases} \quad (4)$$

3 Листинг программы

```
1 import numpy as np
2 from tabulate import tabulate
3
4 A, B = 1, 2
5 TAU = 0.05
6 U10 = 1
7 U20 = -1
8 ORDER = 2
9
10
11 def f1(t, u1, u2):
12     return u2
13
14
15 def f2(t, u1, u2):
16     return -((3 * t + 2) * u2 + u1) / (t * (t + 1))
17
18
19 def u(t):
20     return 1 / t
21
22
23 def du(t):
24     return -1 / (t * t)
25
26
27 def explicit_trapezoid(f1, f2, t, dt, y1_0, y2_0):
28     y1 = np.empty(len(t))
29     y2 = np.empty(len(t))
30     y1[0] = y1_0
31     y2[0] = y2_0
32
33     for i in range(len(t) - 1):
34         _y1 = y1[i] + dt * f1(t[i], y1[i], y2[i])
35         _y2 = y2[i] + dt * f2(t[i], y1[i], y2[i])
36
37         y1[i + 1] = y1[i] + dt / 2 * (f1(t[i], y1[i], y2[i]) + f1(t[i +
38             1], _y1, _y2))
39         y2[i + 1] = y2[i] + dt / 2 * (f2(t[i], y1[i], y2[i]) + f2(t[i +
40             1], _y1, _y2))
41
42     return y1, y2
43
44 def adams(f1, f2, t, dt, y1_0, y2_0):
45     y1 = np.empty(len(t))
46     y2 = np.empty(len(t))
47     y1[0] = y1_0
48     y2[0] = y2_0
49
50     y1[0:2], y2[0:2] = explicit_trapezoid(f1, f2, t[0:2], dt, y1_0, y2_0)
```

```

50
51     for i in range(1, len(t) - 1):
52         y1[i + 1] = y1[i] + dt / 2 * (3 * f1(t[i], y1[i], y2[i]) - f1(t[i]
53             - 1], y1[i - 1], y2[i - 1]))
54         y2[i + 1] = y2[i] + dt / 2 * (3 * f2(t[i], y1[i], y2[i]) - f2(t[i]
55             - 1], y1[i - 1], y2[i - 1]))
56
57     return y1, y2
58
59 def implicit_trapezoid(t, dt, y1_0, y2_0):
60     y1 = np.empty(len(t))
61     y2 = np.empty(len(t))
62     y1[0] = y1_0
63     y2[0] = y2_0
64
65     for i in range(len(t) - 1):
66         a = np.empty((2, 2))
67         a[0][0] = 1
68         a[0][1] = -dt / 2
69         a[1][0] = dt / 2 / (t[i + 1] * (t[i + 1] + 1))
70         a[1][1] = 1 + dt / 2 * (3 * t[i + 1] + 2) / (t[i + 1] * (t[i + 1]
71             + 1))
72
73         b = np.empty(2)
74         b[0] = y1[i] + dt / 2 * y2[i]
75         b[1] = y2[i] - dt / 2 * ((3 * t[i] + 2) * y2[i] + y1[i]) / (t[i] *
76             (t[i] + 1))
77
78         y1[i + 1], y2[i + 1] = np.linalg.solve(a, b)
79
80     return y1, y2
81
82 def max_norm(y1, y2, u1, u2):
83     return np.amax((abs(y1 - u1), abs(y2 - u2)))
84
85 def runge_rule(y1, y2, _y1, _y2, order=2):
86     return np.amax((abs(y1 - _y1[::2]), abs(y2 - _y2[::2]))) / (2 ** order
87         - 1)
88
89 if __name__ == '__main__':
90     t = np.arange(A, B + TAU, TAU)
91     y1_exp_trap, y2_exp_trap = explicit_trapezoid(f1, f2, t, TAU, U10, U20)
92     y1_adams, y2_adams = adams(f1, f2, t, TAU, U10, U20)
93     y1_imp_trap, y2_imp_trap = implicit_trapezoid(t, TAU, U10, U20)
94     u1 = u(t)
95     u2 = du(t)
96
97     tabular_data = {
98         'j': range(len(t)),

```

```

98     't_j': t,
99     'u': u1,
100    'u\\': u2,
101    'y1 expl trap': y1_exp_trap,
102    'y2 expl trap': y2_exp_trap,
103    'y1 adams': y1_adams,
104    'y2 adams': y2_adams,
105    'y1 imp trap': y1_imp_trap,
106    'y2 imp trap': y2_imp_trap,
107 }
108
109 headers_latex = ('j', 't_j', 'u(t_j)', 'u\\'(t_j)', 'y_1(t_j)', 'y_2(
    t_j)',
110                  'y_1(t_j)', 'y_2(t_j)', 'y_1(t_j)', 'y_2(t_j)')
111 floatfmt = ('.1f', '.2f', '.4f', '.4f', '.4f', '.4f', '.4f', '.4f', '
    .4f', '.4f')
112
113 print(tabulate(tabular_data, headers=headers_latex, tablefmt='
    latex_booktabs', floatfmt=floatfmt))
114
115 print(max_norm(y1_exp_trap, y2_exp_trap, u1, u2))
116 print(max_norm(y1_adams, y2_adams, u1, u2))
117 print(max_norm(y1_imp_trap, y2_imp_trap, u1, u2))
118
119 t = np.arange(A, B + TAU / 2, TAU / 2)
120 _y1_exp_trap, _y2_exp_trap = explicit_trapezoid(f1, f2, t, TAU / 2,
    U10, U20)
121 _y1_adams, _y2_adams = adams(f1, f2, t, TAU / 2, U10, U20)
122 _y1_imp_trap, _y2_imp_trap = implicit_trapezoid(t, TAU / 2, U10, U20)
123
124 print(runge_rule(y1_exp_trap, y2_exp_trap, _y1_exp_trap, _y2_exp_trap)
    )
125 print(runge_rule(y1_imp_trap, y2_imp_trap, _y1_imp_trap, _y2_imp_trap)
    )

```

4 Результаты

j	t_j	Точное решение		Явный МТ		ЭМА 2		Неявный МТ	
		$u(t_j)$	$u'(t_j)$	$y_1(t_j)$	$y_2(t_j)$	$y_1(t_j)$	$y_2(t_j)$	$y_1(t_j)$	$y_2(t_j)$
0	1.00	1.0000	-1.0000	1.0000	-1.0000	1.0000	-1.0000	1.0000	-1.0000
1	1.05	0.9524	-0.9070	0.9525	-0.9072	0.9525	-0.9072	0.9523	-0.9068
2	1.10	0.9091	-0.8264	0.9093	-0.8267	0.9095	-0.8276	0.9090	-0.8261
3	1.15	0.8696	-0.7561	0.8698	-0.7565	0.8701	-0.7579	0.8695	-0.7557
4	1.20	0.8333	-0.6944	0.8337	-0.6949	0.8339	-0.6966	0.8332	-0.6939
5	1.25	0.8000	-0.6400	0.8004	-0.6405	0.8006	-0.6425	0.7999	-0.6395
6	1.30	0.7692	-0.5917	0.7696	-0.5922	0.7698	-0.5944	0.7691	-0.5911
7	1.35	0.7407	-0.5487	0.7412	-0.5492	0.7413	-0.5514	0.7406	-0.5481
8	1.40	0.7143	-0.5102	0.7147	-0.5107	0.7148	-0.5130	0.7142	-0.5096
9	1.45	0.6897	-0.4756	0.6901	-0.4761	0.6901	-0.4784	0.6896	-0.4751
10	1.50	0.6667	-0.4444	0.6671	-0.4449	0.6671	-0.4472	0.6666	-0.4439
11	1.55	0.6452	-0.4162	0.6456	-0.4167	0.6455	-0.4189	0.6451	-0.4157
12	1.60	0.6250	-0.3906	0.6254	-0.3911	0.6253	-0.3932	0.6250	-0.3901
13	1.65	0.6061	-0.3673	0.6065	-0.3678	0.6063	-0.3698	0.6060	-0.3668
14	1.70	0.5882	-0.3460	0.5887	-0.3465	0.5884	-0.3484	0.5882	-0.3455
15	1.75	0.5714	-0.3265	0.5718	-0.3270	0.5715	-0.3289	0.5714	-0.3261
16	1.80	0.5556	-0.3086	0.5560	-0.3091	0.5555	-0.3109	0.5556	-0.3082
17	1.85	0.5405	-0.2922	0.5409	-0.2926	0.5404	-0.2943	0.5406	-0.2918
18	1.90	0.5263	-0.2770	0.5267	-0.2774	0.5261	-0.2791	0.5264	-0.2766
19	1.95	0.5128	-0.2630	0.5132	-0.2634	0.5125	-0.2649	0.5129	-0.2626
20	2.00	0.5000	-0.2500	0.5004	-0.2504	0.4996	-0.2519	0.5001	-0.2496

Таблица 1: Численное решение задачи Коши

	Явный МТ	ЭМА	Неявный МТ
Правило Рунге	0.00013	—	0.00014
Максимум норма	0.00050	0.00276	0.00058

Таблица 2: Оценка погрешности для одношаговых методов

5 Выводы

Как видно из Таблицы 2, погрешность явного и неявного методов трапеций одного порядка. Значит, либо шаг был выбран достаточно маленький, либо уравнение не достаточно жесткое