

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский политехнический университет Петра Великого»

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
Высшая школа программной инженерии

Отчет по курсовой работе по дисциплине «Математические модели»

Выполнила студентка гр. 3530904/80001

Прохорова А. И.

Руководитель

Устинов С. М.

Санкт-Петербург
2020

1 Задание

Для **МОДЕЛИ 3** в плоскости параметров (**p4**, **p6**) построить бифуркационные диаграммы точек поворота (**p4**, **p6** > 0). При построении диаграммы целесообразно использовать логарифмический масштаб ($\log(\mathbf{p4}, \mathbf{p6})$) по обеим осям. Убедиться, что это точки поворота, а не ветвления. Проиллюстрировать количество решений в каждой области.
($p_1 = 8.4\text{e-}6$, $p_2 = 6.6667\text{e-}4$, $p_3 = 1.7778\text{e-}5$, $p_5 = 2$).

$$\begin{aligned}\frac{dx_1}{dt} &= (p_1x_2 - x_1x_2 + x_1 - x_1^2)/p_2 - p_4x_1; \\ \frac{dx_2}{dt} &= (-p_1x_2 - x_1x_2 + p_5x_3)/p_3 + p_4(p_6 - x_2); \\ \frac{dx_3}{dt} &= x_1 - x_3 - p_4x_3;\end{aligned}$$

2 Блок аналитических преобразований

Будем решать систему, в качестве ε варьируя p_4

$$\begin{cases} \det(J_N) = 0 \\ f(x, \varepsilon) = 0 \end{cases} \quad (1)$$

Посчитаем матрицу Якоби:

$$J_N = \begin{pmatrix} -\frac{x_2}{p_2} + \frac{1}{p_2} - \frac{2x_1}{p_2 - p_4} & \frac{p_1}{p_2} - \frac{x_1}{p_2} & 0 \\ -\frac{x_2}{p_3} & -\frac{p_1}{p_3} - \frac{x_1}{p_3 - p_4} & \frac{p_5}{p_3} \\ 1 & 0 & -1 - p_4 \end{pmatrix}$$

Запишем систему $f(x, p_4) = 0$:

$$\begin{cases} (p_1x_2 - x_1x_2 + x_1 - x_1^2)/p_2 - p_4x_1 = 0; \\ (-p_1x_2 - x_1x_2 + p_5x_3)/p_3 + p_4(p_6 - x_2) = 0; \\ x_1 - x_3 - p_4x_3 = 0; \end{cases} \quad (2)$$

Из первого уравнения выразим x_2 через x_3 , из третьего уравнения выразим x_3 через x_1 :

$$\begin{aligned}x_2 &= \frac{x_1^2 + p_2p_4x_1 - x_1}{p_1 - x_1} \\ x_3 &= \frac{x_1}{1 + p_4}\end{aligned}$$

Подставим их в уравнение $\det(J_N) = 0$ и получим кубическое уравнение относительно x_1 . Находим корни, подставляем их в выражения для x_2 и x_3 .

p_6 можем найти из второго уравнения:

$$p_6 = \frac{p_1x_2 + x_1x_2 - p_5x_3}{p_3p_4}$$

3 Блок проверки

4 Результаты работы программы

```
p4 = 10.000000
x11 = -0.000036 x12 = 0.000053
x22 = 1.179783 x32 = 0.000005 p6 = 36.582517
check = 0.000000
x23 = 0.587685 x33 = 0.036878 p6 = 92614.526489
check = -0.000000
```

5 Выводы

Для всех вариантов матриц значение обусловленности очень большое (> 50000). Следовательно, доверять полученным результатам мы не можем.

6 Приложение

6.1 Исходный код

```
1 syms x1 x2 x3 p4 p6
2 %double p1 p2 p3 p5
3
4 p1 = 8.4e-6;
5 p2 = 6.6667e-4;
6 p3 = 1.7778e-5;
7 p5 = 2;
8
9 x2val = (x1 * x1 + p2 * p4 * x1 - x1) / (p1 - x1);
10 x3val = x1 / (1 + p4);
11 p6 = (p1 * x2 + x1 * x2 - p5 * x3) / p3 * p4 + x2;
12
13 det11 = -x2/p2 + 1/p2 - 2*x1/p2 - p4;
14 det12 = p1 / p2 - x1 / p2;
15 det13 = 0;
16 det21 = -x2/p3;
17 det22 = - p1 / p3 - x1 / p3 - p4;
18 det23 = p5/p3;
19 det31 = 1;
20 det32 = 0;
21 det33 = - 1 - p4;
22
23 det11 = subs(det11, x2, x2val);
24 det21 = subs(det21, x2, x2val);
25
26 res = det ( [det11 det12 det13; det21 det22 det23; det31 det32 det33] );
27 res2 = det ( [det11 det12 det13; det21 det22 det23; det31 det32 det33] );
28
29 eq = res2 == 0;
30
31 fileID = fopen('resulst.txt', 'w');
32 fileIDcheck = fopen('check.txt', 'w');
33
34 %for i = 1.0: 0.1: 2.0
35 for i = 10.0: 1.0: 10.0
36     %p4 = i;
37     inner = subs(eq, p4, i);
38     x2tmp = subs(x2val, p4, i);
39     x3tmp = subs(x3val, p4, i);
40     p6tmp = subs(p6, p4, i);
41     check = subs(res, p4, i);
42
43     fprintf(fileID, 'p4 = %f \n', i);
44     fprintf(fileIDcheck, '%f\n', i);
45
46     %solution = sym(solve(inner, x1,'IgnoreProperties', true));
47     %solution = solve(res, x1);
48     solution = solve(inner, x1);
49     if (isempty(solution)==false)
50         x11 = solution(1, 1);
51         x12 = solution(2, 1);
52         x13 = solution(3, 1);
```

```

53
54
55 fprintf(fileID, 'x11 = %f x12 = %f \n', x11, x12);
56 if (isAlways(x11 > 0) == true)
57     x21 = subs(x2tmp, x1, x11);
58     x31 = subs(x3tmp, x1, x11);
59     p6res = subs(p6tmp, x1, x11);
60     p6res = subs(p6res, x2, x21);
61     p6res = subs(p6res, x3, x31);
62     fprintf(fileID, 'x21 = %f x31 = %f p6 = %f \n', x21, x31, p6res);
63     p6check = (8.4e-6 * x21 + x11 * x21 - 2 * x31) / 1.7778e-5 * i + x21;
64
65     check1 = subs(check, x1, x11);
66     fprintf(fileIDcheck, '%f %f %f %f\n', x11, x21, x31, p6res);
67
68 end
69 if (isAlways(x12 > 0))
70     x22 = subs(x2tmp, x1, x12);
71     x32 = subs(x3tmp, x1, x12);
72     p6res = subs(p6tmp, x1, x12);
73     p6res = subs(p6res, x2, x22);
74     p6res = subs(p6res, x3, x32);
75     fprintf(fileID, 'x22 = %f x32 = %f p6 = %f \n', x22, x32, p6res);
76     p6check = (8.4e-6 * x22 + x12 * x22 - 2 * x32) / 1.7778e-5 * i + x22;
77
78     check2 = subs(check, x1, x12);
79     fprintf(fileID, 'check = %f \n', check2);
80     fprintf(fileIDcheck, '%f %f %f %f\n', x12, x22, x32, p6res);
81 end
82 if (isAlways(x13 > 0) == true)
83     x23 = subs(x2tmp, x1, x13);
84     x33 = subs(x3tmp, x1, x13);
85     p6res = subs(p6tmp, x1, x13);
86     p6res = subs(p6res, x2, x23);
87     p6res = subs(p6res, x3, x33);
88     fprintf(fileID, 'x23 = %f x33 = %f p6 = %f \n', x23, x33, p6res);
89     p6check = (8.4e-6 * x23 + x13 * x23 - 2 * x33) / 1.7778e-5 * i + x23;
90
91     check3 = subs(check, x1, x13);
92     fprintf(fileID, 'check = %f \n', check3);
93     fprintf(fileIDcheck, '%f %f %f %f\n', x13, x23, x33, p6res);
94 end
95 fprintf(fileID, '\n');
96 end
97
98
99 end
100 fclose(fileID);

```