

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Сибирский государственный университет телекоммуникаций и
информатики»
(СибГУТИ)

Кафедра вычислительных систем

ОТЧЕТ
по курсовой работе
по дисциплине «**Вычислительная математика**»

Выполнил:
студент гр. ИС-242
«15» мая 2024 г.

/Пухов М. С./

Проверил:
преподаватель
«__» мая 2024 г.

/Чупрыно Л. А./

Оценка «_____»

Новосибирск 2024

Оглавление

ПОСТАНОВКА ЗАДАЧИ	3
МЕТОД РЕШЕНИЯ.....	5
СХЕМА АЛГОРИТМА РЕШЕНИЯ.....	6
ЛИСТИНГ ПРОГРАММЫ	7
РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ	9

ПОСТАНОВКА ЗАДАЧИ

В рамках данной курсовой требуется решить систему уравнений SEIR-D, моделирующую распространение инфекции COVID-19 в Новосибирской области. Система уравнений вместе с коэффициентами и начальными данными представлены в статье сибирского журнала вычислительной математики «Математическое моделирование и прогнозирование COVID-19 в Москве и Новосибирской области». Решение должно быть найдено с помощью метода Эйлера на интервале от 0 до 90 дней с точностью до 2 знаков после запятой.

Система уравнений (из статьи):

2. Математическая модель SEIR-D

В рамках модели SEIR-D распространение коронавируса COVID-19 описывается системой из 5 нелинейных обыкновенных дифференциальных уравнений на отрезке $t \in [t_0, T]$ [31] (схема модели приведена на рис. 1 справа):

$$\begin{cases} \frac{dS}{dt} = -c(t - \tau) \left(\frac{\alpha_I S(t) I(t)}{N} + \frac{\alpha_E S(t) E(t)}{N} \right) + \gamma R(t), \\ \frac{dE}{dt} = c(t - \tau) \left(\frac{\alpha_I S(t) I(t)}{N} + \frac{\alpha_E S(t) E(t)}{N} \right) - (\kappa + \rho) E(t), \\ \frac{dI}{dt} = \kappa E(t) - \beta I(t) - \mu I(t), \\ \frac{dR}{dt} = \beta I(t) + \rho E(t) - \gamma R(t), \\ \frac{dD}{dt} = \mu I(t). \end{cases} \quad (5)$$

Здесь $N = S + E + I + R + D$ — вся популяция.

Функция, использующая ограничения на передвижения граждан:

$$c(t) = 1 + c^{\text{isol}} \left(1 - \frac{2}{5} a(t) \right), \quad c(t) \in (0, 2).$$

Начальные данные:

$$S(t_0) = S_0, \quad E(t_0) = E_0, \quad I(t_0) = I_0, \quad R(t_0) = R_0, \quad D(t_0) = D_0. \quad (6)$$

Параметры для периода измерений (из статьи):

Таблица 11. Восстановленные параметры для периода измерений 23.03.2020–31.05.2020, Новосибирская область

Модель	α_E	α_I	κ	ρ	β	ν	ε_{CH}	μ	c^{isol}	E_0	R_0
SEIR-HCD	0.001	0.224	0.108	–	0.013	0.006	0.055	0.072	–	1001	–
SEIR-D	0.999	0.999	0.042	0.952	0.999	–	–	0.0188	0	99	24

Начальные данные о распространении (из статьи):

3.4. Численные расчеты для Новосибирской области

В качестве начальных данных о распространении коронавируса COVID-19 в Новосибирской области ($N_0 = 2\,798\,170$ человек) для математической модели SEIR-HCD была взята статистическая информация за 23 марта 2020 г.:

$$S_0 = 2\,798\,170 - q_8, \quad E_0 = q_8, \quad I_0 = 0, \quad R_0 = 0, \quad H_0 = 0, \quad C_0 = 0, \quad D_0 = 0,$$

а для математической модели SEIR-D в следующем виде:

$$S_0 = 2\,798\,170 - q_8 - q_9, \quad E_0 = q_8, \quad I_0 = 0, \quad R_0 = q_9, \quad D_0 = 0.$$

Решенная задача (из статьи)



а) Измерения по 31.05.2020.

Параметры приведены в табл. 11

График №2.

МЕТОД РЕШЕНИЯ

Постановка задачи предполагает решение системы с точностью до 2 знаков после запятой методом Эйлера, но так как классический метод Эйлера обладает недостаточной точностью для выполнения задачи (метод первого порядка точности), в решении использовался усовершенствованный метод Эйлера или же метод Эйлера-Коши, который является методом второго порядка точности и позволяет получить решение с требуемой точностью.

Метод Эйлера-Коши - это численный метод, используемый для решения обыкновенных дифференциальных уравнений (ОДУ) первого порядка. Он основан на идее аппроксимации решения дифференциального уравнения с помощью приращений функции в точке, используя значения функции и её производной в этой точке.

Метод Эйлера-Коши позволяет решать уравнения вида:

$$\begin{cases} y' = f(x, y), \\ y(x_0) = y_0. \end{cases}$$

Решение задачи Коши выглядит следующим образом:

$$\begin{cases} \tilde{y}_{i+1} = y_i + h f(x_i, y_i), \\ y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1})] \end{cases}$$

При данном методе решение ищется на отрезке $[x_0, b]$ с разбиением отрезка на n равных частей. При таком подходе $h = \frac{b-x_0}{n}$, $x = x_0 + hi$, $i = 0, 1, \dots, n$.

СХЕМА АЛГОРИТМА РЕШЕНИЯ

Алгоритм реализует модель SEIRD, которая используется для моделирования распространения инфекционных заболеваний, таких как эпидемия COVID-19. Описание самого алгоритма:

1. **Инициализация переменных:** Задание начальных значений для всех переменных, таких как численность населения, параметры модели (aI , aE , K , P , m , B), начальные значения для S , E , I , R , D и другие.
2. **Цикл расчета:** Используя метод Эйлера-Коши, вычисляем значения S , E , I , R , D на каждом временном шаге. В этом цикле происходят следующие шаги:
 - Рассчитывается текущая численность населения N .
 - Рассчитывается шаг времени h .
 - Вычисляются приращения S , E , I , R , D , используя функции `susceptible`, `ezaraz`, `infected`, `recovered`, `dead`.
 - Происходит обновление значений S , E , I , R , D на текущем временном шаге.
3. **Запись результатов:** Записываем результаты в файл для последующего анализа. Каждая строка файла содержит номер дня и соответствующее значение функции `group_SEIRD`.

Этот алгоритм позволяет моделировать количество выявленных случаев заражения.

ЛИСТИНГ ПРОГРАММЫ

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  double susceptible(double aI, double S, double I, double N, double aE,
6  double E)
7  {
8      return (-1 * ((aI * S * I) / N) + ((aE * S * E) / N));
9  }
10
11 double ezaraz(double aI, double S, double I, double N, double aE, double
12 E, double K, double P)
13 {
14     return (((aI * S * I) / N) + ((aE * S * E) / N)) - ((K + P) * E);
15 }
16
17 double infected(double K, double E, double I, double m, double B)
18 {
19     return (K * E - B * I - m * I);
20 }
21
22 double recovered(double B, double I, double P, double E)
23 {
24     return (B * I + P * E);
25 }
26
27 double dead(double m, double I)
28 {
29     return (m * I);
30 }
31
32 double group_SEIRD(double S_prev, double E_prev, double I_prev, double
33 R_prev, double D_prev, double day)
34 {
35     double aI = 0.999;
36     double aE = 0.999;
37     double K = 0.042;
38     double P = 0.952;
39     double m = 0.0188;
40     double B = 0.999;
41
42     double h = day/90;
43
44     int N;
45     for (int i = 0; i < day; i++)
46     {
47         double S2, E2, I2, R2, D2;
48
49         double S1, E1, I1;
50
51         N = S_prev + E_prev + I_prev + R_prev + D_prev;
52
53         S1 = S_prev + (h * susceptible(aI, S_prev, I_prev, N, aE,
54 E_prev));
55         E1 = E_prev + (h * ezaraz(aI, S_prev, I_prev, N, aE, E_prev, K,
P));
```

```

56         I1 = I_prev + (h * infected(K, E_prev, I_prev, m, B));
57
58         S2 = S_prev + (h / 2 * (susceptible(aI, S_prev, I_prev, N, aE,
59 E_prev) + susceptible(aI, S1, I1, N, aE, E1)));
60         E2 = E_prev + (h / 2 * (ezaraz(aI, S_prev, I_prev, N, aE,
61 E_prev, K, P) + ezaraz(aI, S1, I1, N, aE, E1, K, P)));
62         I2 = I_prev + (h / 2 * (infected(K, E_prev, I_prev, m, B) +
63 infected(K, E1, I1, m, B)));
64         R2 = R_prev + (h / 2 * (recovered(B, I_prev, P, E_prev) +
65 recovered(B, I1, P, E1)));
66         D2 = D_prev + (h / 2 * (dead(m, I_prev) + dead(m, I1)));
67
68         S_prev = S2;
69         E_prev = E2;
70         I_prev = I2;
71         R_prev = R2;
72         D_prev = D2;
73     }
74
75     return (K * E_prev) / 0.58;
76 }
77
78 int main()
79 {
80
81     double ezaraz = 99;
82     double infected = 0;
83     double recovered = 24;
84     double dead = 0;
85     double susceptible = 2798170 - ezaraz - recovered;
86
87     FILE *fd = fopen("1.txt", "w");
88     if (fd == NULL)
89     {
90         printf("Error open file\n");
91         exit(EXIT_FAILURE);
92     }
93
94     char buf[100];
95     for (int i = 1; i < 91; i++)
96     {
97         double Fk = group_SEIRD(susceptible, ezaraz, infected,
recovered, dead, i);
98         sprintf(buf, "%d %.6f\n", i, Fk);
99         fwrite(buf, sizeof(char), strlen(buf), fd);
100     }
101
102     fclose(fd);
103 }

```


РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

Результатом работы программы является график, построенный на основе данных, посчитанных с помощью программы. (рис.1.)

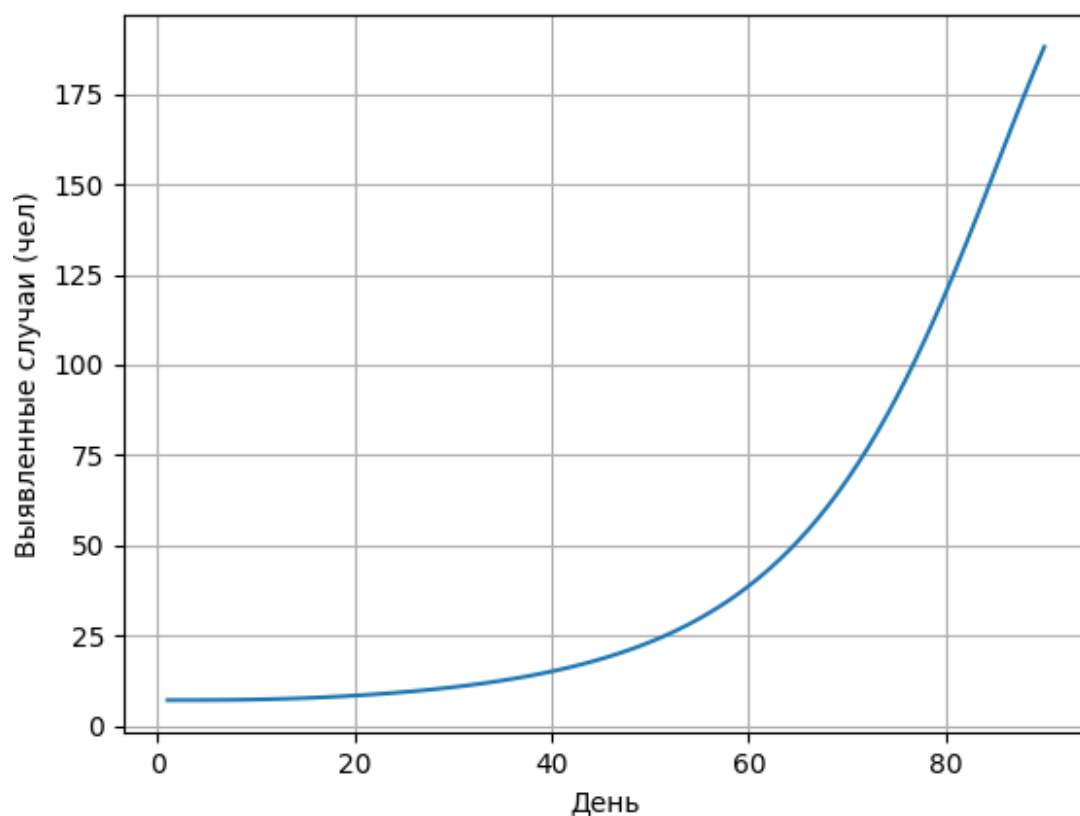


Рисунок 1. Количество выявленных случаев заражения на интервале от 0 до 90 дней.