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](#_bookmark0)

[МЕТОД РЕШЕНИЯ 5](#_bookmark1)

[СХЕМА АЛГОРИТМА РЕШЕНИЯ 6](#_bookmark2)

[ЛИСТИНГ ПРОГРАММЫ 7](#_bookmark3)

[РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ 9](#_bookmark4)

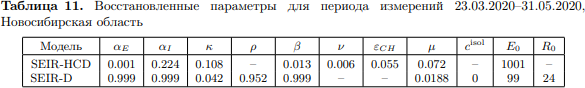
# ПОСТАНОВКА ЗАДАЧИ

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.

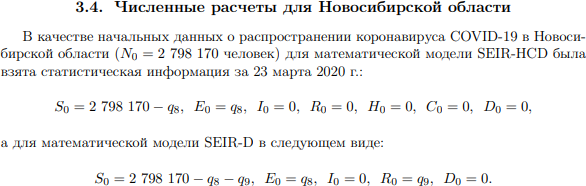
Система уравнений (из статьи):



Параметры для периода измерений (из статьи):



Начальные данные о распространении (из статьи):



Решенная задача (из статьи)

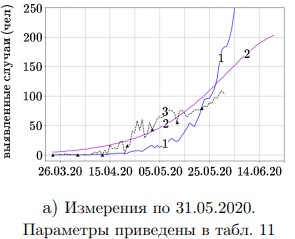


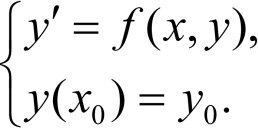
График №2.

# МЕТОД РЕШЕНИЯ

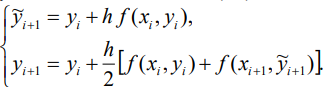
Постановка задачи предполагает решение системы с точностью до 2 знаков после запятой методом Эйлера, но так как классический метод Эйлера обладает недостаточной точностью для выполнения задачи (метод первого порядка точности), в решении использовался усовершенствованный метод Эйлера или же метод Эйлера-Коши,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



Решение задачи Коши выглядит следующим образом:



При данном методе решение ищется на отрезке [x0, b] с разбиением

отрезка на n равных частей. При таком подходе ℎ = 𝑏−𝑥0, 𝑥 = 𝑥

+ ℎ𝑖, i = 0, 1,

𝑛 0

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

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

# РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

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

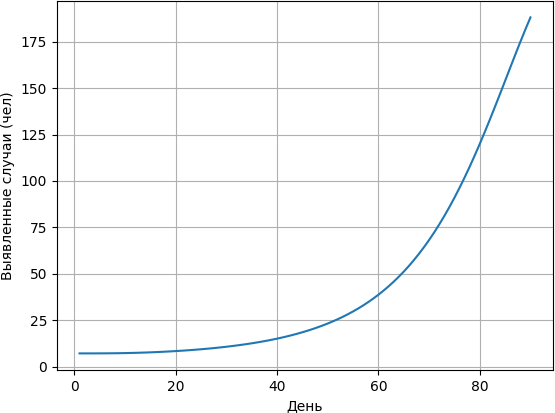


Рисунок 1. Количество выявленных случаев заражения на интервале от 0 до 90 дней.