Групповой проект номер №5

Дисциплина "Научное программирование" Этап Второй

Супонина Анастасия Павловна, Лобов Михаил Сергеевич, Нирдоши Всеволод Раджендер

05 Октября 2024

Кафедра теории вероятностей и кибербезопасности

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Преимущества дифференциальных уравнений для решения задач

1. **Моделирование сложных систем**: Дифференциальные уравнения необходимы для моделирования реальных систем, таких как эпидемии (модель SIR), где изменения в популяциях (например, зараженные, восприимчивые, выздоровевшие) происходят во времени.
2. **Непрерывные решения**: Они предоставляют непрерывные решения, что позволяет понять изменения в любой момент времени.
3. **Гибкость**: Дифференциальные уравнения позволяют интегрировать различные параметры и условия (например, коэффициенты заражения и выздоровления) и адаптироваться к динамическим изменениям (например, пороги эпидемий).

Преимущества дифференциальных уравнений для решения задач

4. **Прогнозирование**: Решая дифференциальные уравнения, можно предсказать поведение системы, например, время до достижения равновесия или прекращения эпидемии.
5. **Численные методы**: Для сложных систем, где аналитические решения невозможны, численные методы (например, метод Рунге-Кутты) позволяют получать приближенные решения эффективно.

Алгоритм решения для случая I <= I*. Переменные.

Для данного случая из модели SIR мы работаем со следующими переменными:

t = 0 - начальное время

I = 700 - количество инфицированных особей, распространители инфекции

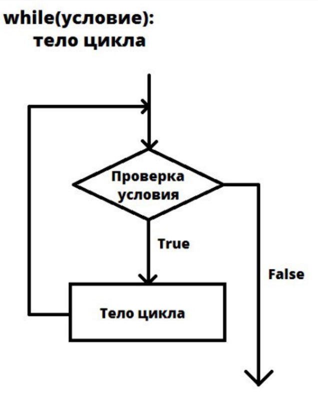
R = 0 - количество здоровых особей с иммунитетом к болезни

a = 0.007 - коэффициент заболевания

b = 0.003 - коэффициент выздоровления

Алгоритм решения для случая I <= I*. Цикл While.

1. Так как мы не знаем точное количество дней для окончания эпидемии, то будем использовать цикл while.
2. Так как мы будем брать счетчик для времени с запасом, то нам необходимо дополнительное условие для выхода из цикла. Будем считать что, когда количество распространителей(I) станет равно 0 эпидемия закончится.



Алгоритм решения для случая I <= I*. Тело цикла.

3. Внутри цикла будем записывать наши дифференциальный уравнения из модели SIR, для этого случая только с I и R
4. Также добавим счетчик, чтобы знать количество дней окончания эпидемии (t + 1)
5. Для построения графика нам нужно создать пустые списки, куда будут записываться переменные на каждом шаге
   count_I = [ ] - счетчик для I
   count_R = [ ] - счетчик для R

$$\frac{dI}{dt} = -\beta I \qquad \frac{dR}{dt} = \beta I$$

Алгоритм решения для случая I > I*.

К переменным из прошлой задачи добавляем:

1. N - общее количество особей в популяции, которое нужно для вычисления S
2. S = N-I-R - количество здоровых особей, но восприимчивых к болезни
3. count_S = [] - счетчик для S

Учитывая новые переменные в программу добавляется дифференциальное уравнение из модели SIR и счетчик для S, а также меняются уравнения для I и R.

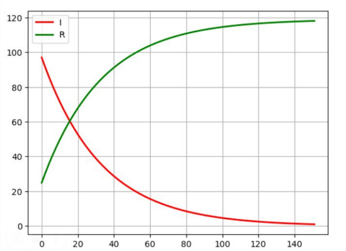
$$\frac{dS}{dt} = -\alpha S \quad \frac{dI}{dt} = aS - \beta I \quad \frac{dR}{dt} = \beta I$$

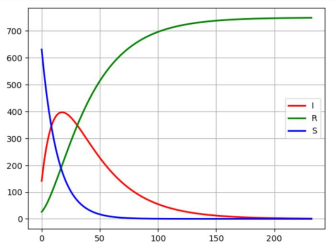
Ограничения и цикл while не меняется!

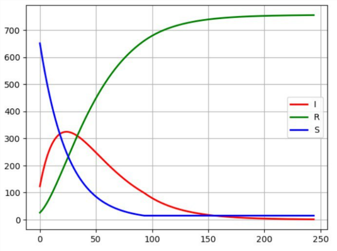
Обобщенное решение.

Мы объединяем два случая для получения полной картины модели об эпидемии. (Т.к. в реальности число зараженный может упасть ниже критического значения, даже если оно было выше ранее) С помощью условия If внутри цикла раскроем сразу 2 случая: когда число зараженных превышает и не превышает критический порог В зависимости от динамического выполнения условия будет изменяться ход цикла Как условие будем использовать I > I*

После выполнения данных комманд мы должны будем получить следующие графики.







Метод Рунге-Кутты 4го порядка

Метод Рунге-Кутты - один из самых широко используемых численных методов для решения ОДУ. Он был разработан Карлом Рунге и Мартином Куттой в начале XX века и является усовершенствованием метода Эйлера. Основная цель метода Рунге-Кутты — аппроксимировать решение дифференциального уравнения, шаг за шагом вычисляя значения искомой функции на определённом интервале Алгоритм 4-го порядка включает следующие шаги:

1. **Первый уклон (начало шага):**

$$d_1 = f(t, y)$$

Для каждой переменной (S, I, R) Это уклон в начале интервала.

2. **Второй уклон (середина шага):**

$$d_2 = f\left(t + \frac{\Delta t}{2}, y + \frac{k_1 \Delta t}{2}\right)$$

Для каждой переменной (S, I, R) Это уклон, вычисленный в середине интервала с использованием аппроксимации на основе k_1.

3. **Третий уклон (ещё одна середина шага):**

$$d_3 = f\left(t + \frac{\Delta t}{2}, y + \frac{k_2 \Delta t}{2}\right)$$

Для каждой переменной (S, I, R) Это ещё один уклон, вычисленный также в середине интервала, но с использованием уже k_2.

4. **Четвёртый уклон (конец шага):**

$$d_4 = f(t + \Delta t, y + k_3 \Delta t)$$

Для каждой переменной (S, I, R) Это уклон на конце интервала, аппроксимированный на основе предыдущих значений.

Метод Рунге-Кутты 4го порядка

Метод 4-го порядка даёт точное решение, поскольку учитывает несколько промежуточных значений уклонов, что позволяет лучше аппроксимировать динамику заражения и выздоровления.

В дальнейшем мы сравним при реализации эти два метода решения.

1. **Библиотека `DifferentialEquations.jl` в Julia**: Julia предоставляет мощную библиотеку для решения обыкновенных дифференциальных уравнений (ODE) с помощью `DifferentialEquations.jl`.

- **Простая настройка задачи**: С помощью функции `ODEProblem` можно задать дифференциальное уравнение, начальные условия и временной промежуток для решения.

- **Эффективные решатели**: Julia поддерживает различные численные методы, включая методы Рунге-Кутты, которые позволяют эффективно решать сложные и масштабные системы.

2. **Графический вывод**: Решение можно легко визуализировать на графике, чтобы увидеть, как изменяются популяции во времени (например, восприимчивые, зараженные, выздоровевшие).

Спасибо за внимание!