Реферат по теме: «Модель заражения SIS»

по дисциплине: Математическое моделирование

Ким Михаил Алексеевич

Содержание

Список иллюстраций

1 Цель и задачи работы

Изучить математическую модель заражения SIS. Используя функционал языка программирования Julia вместе с дополнительными библиотеками (DifferentialEquations, Plots), языка моделирования Modelica, а также интерактивного блокнота Pluto и программного обеспечением OpenModelica, описать модель заражения SIS. Сравнить описанную математическую модель с реальными данными о заражении.

2 Теоретическое введение

Эпидемии издавна являлись большой угрозой для человечества. В XXI веке мир уже успел столкнуться с эпидемией птичьего гриппа в Юго-Восточной Азии (в 2013 году), вспышкой заболеваний лихорадкой Эбола в Африке (2015), пандемией COVID-19, начавшейся в 2019 году и продолжающейся до сих пор. Но в истории человечества бывали и куда более масштабные эпидемии.

В конце XI нашей эры в Римской империи разразилась первая задокументированная пандемия чумы, в результате которой погибло около 100 миллионов человек. Спустя еще XII веков в Евразию и Северную Африку пришла Черная смерть — пандемия чумы, сразившая от трети до половины тогдашнего населения этих регионов.

В результате Первой мировой войны, вызвавшей перемещение большого количества людей, в 1918 году распространился испанский грипп, охвативший более 500 миллионов человек и погубивший каждого десятого заболевшего. Это далеко не все случаи возникновения эпидемий, погубивших в конечном счете бесчисленное количество невинных жизней.

Только в XX веке были разработаны эффектинвые средства борьбы с инфекциями. К числу этих средств принадлежат и системы дифференциальных уравнений — математика помогает моделировать распространение эпидемий и помогает понять, как следует с ними бороться. Изучение механизмов развития и распространения эпидемий является важным способом борьбы с заболеваниями наряду с поиском новых лекарств, вакцинацией и профилактическими мерами [1].

Наряду с моделью SIS при описании распространения инфекций используется целый ряд других моделей, к примеру, SIR, SEIR, MSEIR [2] и др. Более того, эпидимологическую модель SIS можно считать последующим развитием модели SIR. Но в рамках данного реферата мы остановимся конкретно на упомянутой ранее модели «Susceptible — Infected — Susceptible» — SIS.

Как следует из расшифровки аббревиатуры, модель SIS включает в себя две группы объектов: Susceptible (восприимчивые — еще не инфицированные организмы, которые, однако, могут быть подвержены заражению), Infected (инфицированные — заразившиеся организмы) [3].

Также, все еще благодаря расшифровке аббревиатуры, мы можем отследить последовательность перехода объектов из одной группы в другую: восприимчивые становятся инфицированными, и после выздоровления снова становятся восприимчивыми. Такая последовательность перехода и определяет множество инфекций, в которых применима модель: к примеру, грипп и ОРВИ (заболевания, к которым не вырабатывается иммунитет) [4].

Модель описывается слудеющей системой уравнений:

$$\frac{dS}{dt} = -\frac{\beta SI}{N} + \gamma I$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I,$$

где $S(t)$ — численность восприимчивых (susceptible) индивидов в момент времени t, $I(t)$ — численность инфицированных (infected) индивидов в момент времени t, β — коэффициент интенсивности контактов индивидов с последующим инфицированием, γ — коэффициент интенсивности выздоровления инфицированных индивидов, N — число объектов в популяции.

Первое уравнение описывает изменение числа восприимчивых в единицу времени, которое уменьшается на число зараженных (первое слагаемое) и увеличивается на число выздоровеших (второе слагаемое).

Рассмотрим первое слагаемое подробно. $\frac{\beta SI}{N}$ можно представить в виде:

$$\frac{1}{N} \cdot \beta SI = \frac{\beta}{N} \cdot SI = \frac{\beta S}{N} \cdot I = \frac{\beta SI}{N}$$

где $\frac{1}{N}$ — вероятность контакта между двумя индивидами (подразумевается, что в каждый момент времени каждый индивид контактирует с одним случайным индивидом в популяции), $\frac{\beta}{N}$ — вероятность контакта и заражения между двумя индивидами, $\frac{\beta S}{N}$ — суммарное число зараженных индивидов инфицированным, $\frac{\beta SI}{N}$ — суммарное число зараженных индивидом всеми инфицированными.

Рассмотрим второе слагаемое подробно: каждый инфицированный в определенный момент времени может выздороветь с вероятностью γ. Общее число выздоровевших инфицированных в определенный момент времени есть $\gamma \cdot I$.

Второе уравнение характеризует изменение числа заболевших в единицу времени, которое пропорционально числу заражений (числу контактов здоровых и инфицированных индивидуумов) за вычетом числа выздоровлений. Все слагаемые данного уравнения подробно описаны выше.

Величина $R_0 = \frac{\beta}{\gamma}$ является «базовым коэффициентом воспроизведения» и имеет большую значимость при оценке возможности распространения болезни (чем он больше, тем более болезнь заразна). К примеру, у COVID-19 $R_0 = 2.4 - 3.4$, у кори $R_0 = 12 - 18$, у гриппа $R_0 = 0.9 - 2.1$ [5] [6] [7].

Важно также отметить, что справедливы следующие уравнения:

$$\frac{dS}{dt} + \frac{dI}{dt} = 0 \;\Rightarrow\; S(t) + I(t) = N$$

Из правого уравнения следует, что суммарное число восприимчивых и инфицированных всегда остается одинаковым и равным N. Соответственно, стандартная модель SIS предполагает, что в популяции отсутствует рождаемость и смертность от болезни [8].

3 Построение модели

3.1 Pluto.jl

Пишем программу, воспроизводящую модель на языке программирования Julia с использованием интерактивного блокнота Pluto.

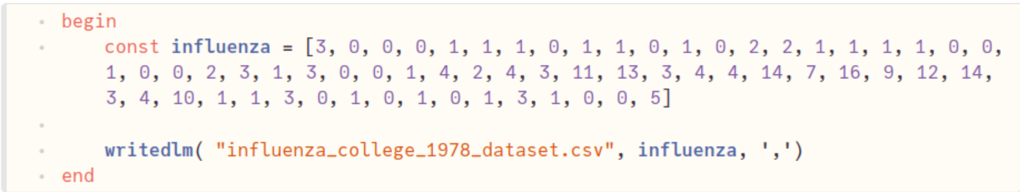
1. Импорт необходимых библиотек (рис. 3.1).

```
begin
    import Pkg
    Pkg.activate()
    using DifferentialEquations
    using LaTeXStrings
    import Plots
end
```

Рис. 3.1: Импорт необходимых библиотек

2. Формирование датасета и запись его в файл формата .csv (рис. 3.2).

```
begin
    const influenza = [3, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 2, 2, 1, 1, 1,

    writedlm( "influenza_college_1978_dataset.csv", influenza, ',')
end
```



```
begin
    const influenza = [3, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 2, 2, 1, 1, 1, 1, 0, 0,
    1, 0, 0, 2, 3, 1, 3, 0, 0, 1, 4, 2, 4, 3, 11, 13, 3, 4, 4, 14, 7, 16, 9, 12, 14,
    3, 4, 10, 1, 1, 3, 0, 1, 0, 1, 0, 1, 3, 1, 0, 0, 5]

    writedlm( "influenza_college_1978_dataset.csv", influenza, ',')
end
```

Рис. 3.2: Формирование датасета и запись его в файл

3. Импорт датасета из файла. Задание вектора времени (рис. 3.3).

```
begin
    df = readdlm("influenza_college_1978_dataset.csv", ',', Int64)
    const days = [i for i in 1:length(df)]
    @show df
end
```

```
63×1 Matrix{Float64}:
 3.0
 0.0
 0.0
 0.0
 1.0
 1.0
 1.0
 ⋮
 1.0
 3.0
 1.0
 0.0
 0.0
 5.0
```

```
begin
    df = readdlm("influenza_college_1978_dataset.csv", ',', Float64)
    const days = [i for i in 1:length(df)]
    @show df
end
```

```
df = [3.0; 0.0; 0.0; 0.0; 1.0; 1.0; 1.0; 0.0; 1.0; 1.0; 0.0; 1.0; 0.0; 2.0;
2.0; 1.0; 1.0; 1.0; 1.0; 0.0; 0.0; 1.0; 0.0; 0.0; 2.0; 3.0; 1.0; 3.0; 0.0; 0.0;
1.0; 4.0; 2.0; 4.0; 3.0; 11.0; 13.0; 3.0; 4.0; 4.0; 14.0; 7.0; 16.0; 9.0; 12.0;
14.0; 3.0; 4.0; 10.0; 1.0; 1.0; 3.0; 0.0; 1.0; 0.0; 1.0; 0.0; 1.0; 3.0; 1.0; 0.
0; 0.0; 5.0;;]
```

Рис. 3.3: Импорт датасета из файла. Задание вектора времени

4. Преобразование датасета (рис. 3.4).

```
begin
    df_sis = zeros(Int64, size(df)[1])
    df_sis[1] = df[1]
    for i = 2:length(df)
        df_sis[i] += df_sis[i-1] + df[i]
    end
    @show df_sis
end
```

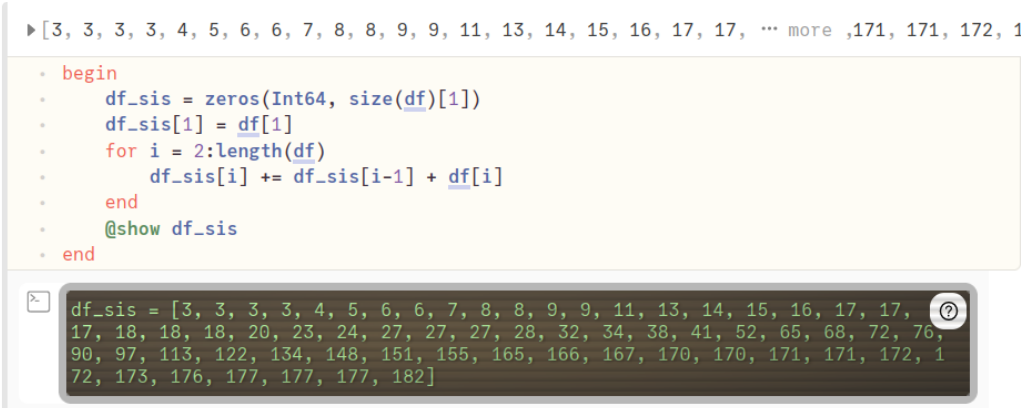


Рис. 3.4: Преобразование датасета

5. Сравнение графика источника (рис. 3.5) с построенным графиком (рис. 3.6). Построение графика по обработанному датасету (рис. 3.7).

```
begin
    fig1 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Случаи заболевания гриппом в кампусе колледжа в 1978"
    )

    Plots.plot!(
        fig1[1],
        days,
        df,
        color=[:red],
        xlabel="Число дней",
```

```
        ylabel="Число заболеваний в день",
        label="Число зафиксированных заболеваний за день"
    )
end
```

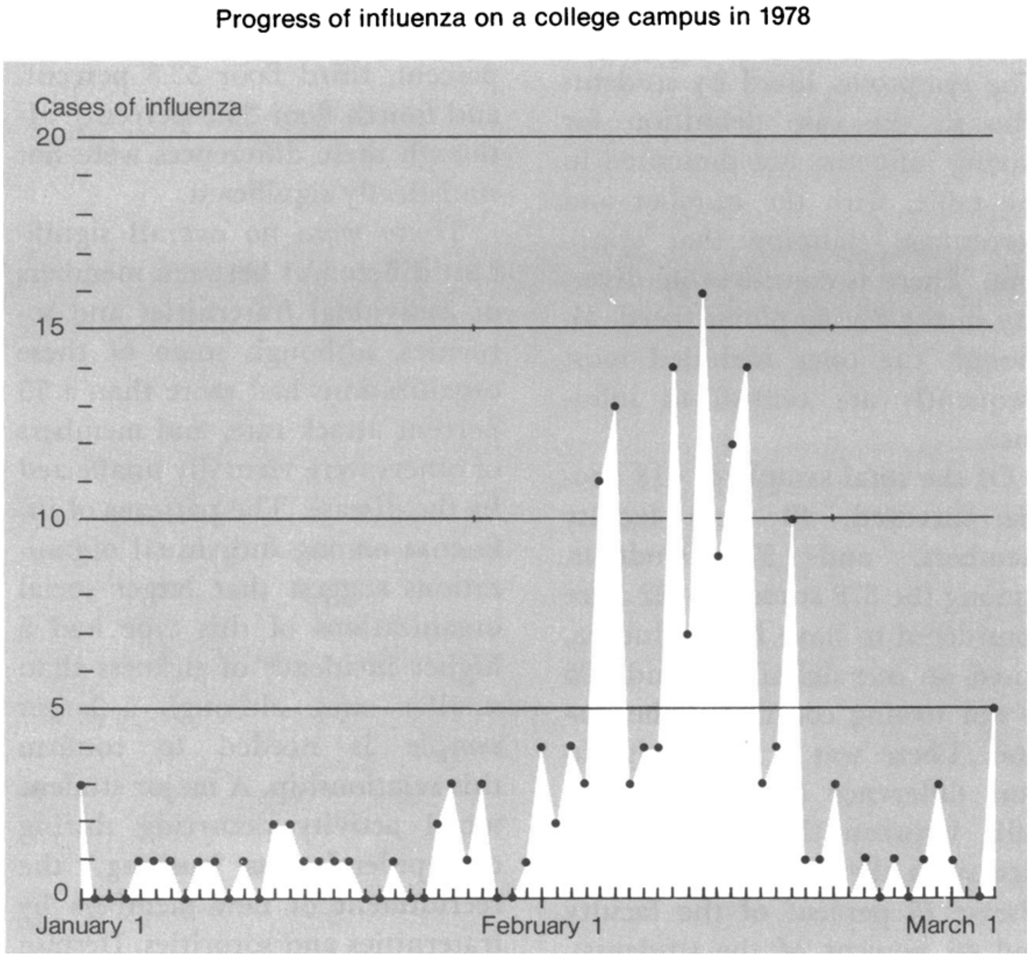


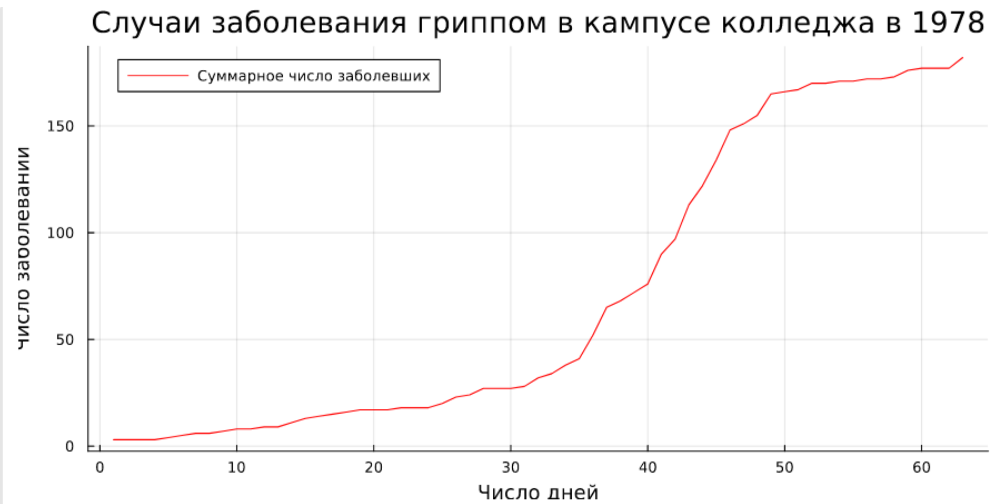
Рис. 3.5: График источника


```
begin
    fig1 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Случаи заболевания гриппом в кампусе колледжа в 1978"
    )

    Plots.plot!(
        fig1[1],
        days,
        df,
        color=[:red],
        xlabel="Число дней",
        ylabel="Число заболеваний в день",
        label="Число зафиксированных заболеваний за день"
    )
end
```

Рис. 3.6: Необработанный график созданного датасета

```
begin
    fig2 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
```

```
        plot_title="Случаи заболевания гриппом в кампусе колледжа в 1978"
    )

    Plots.plot!(
        fig2[1],
        days,
        df_sis,
        color=[:red],
        xlabel="Число дней",
        ylabel="Число заболеваний",
        label="Суммарное число заболевших"
    )
end
```




```
begin
    fig2 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Случаи заболевания гриппом в кампусе колледжа в 1978"
    )

    Plots.plot!(
        fig2[1],
        days,
        df_sis,
        color=[:red],
        xlabel="Число дней",
        ylabel="Число заболеваний",
        label="Суммарное число заболевших"
    )
end
```

Рис. 3.7: Обработанный график созданного датасета

6. Описание модели SIS. Блок параметров. $R_0 = 1.875$ (рис. 3.8).

```
begin
    const ⊠ = 0.3
    const ⊠ = 0.16
    @show R⊠ = ⊠ / ⊠

    const N = 418
    const I⊠ = 1
```

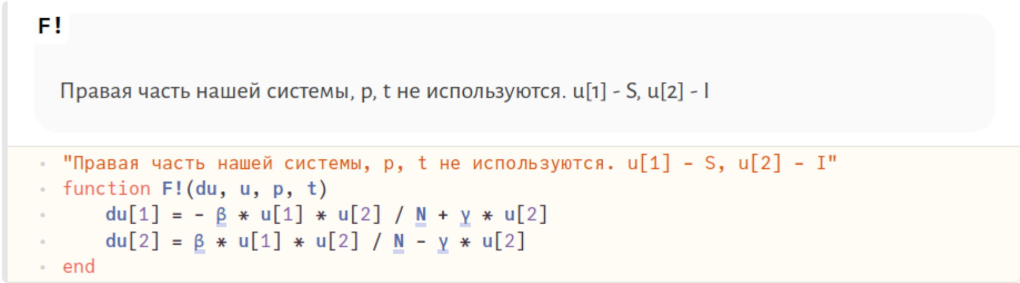
```
    const S⊠ = N - I⊠
    @show S⊠

    "Начальные условия: u⊠[1] - S⊠, u⊠[1] - I⊠"
    u⊠ = [S⊠, I⊠]

    "Период времени"
    T = (1.0, length(df))
end
```

Рис. 3.8: Блок параметров

7. Описание модели SIS. Функция, задающая систему ОДУ (рис. 3.9).

```
"Правая часть нашей системы, p, t не используются. u[1] - S, u[2] - I"
function F!(du, u, p, t)
    du[1] = - ⊠ * u[1] * u[2] / N + ⊠ * u[2]
    du[2] = ⊠ * u[1] * u[2] / N - ⊠ * u[2]
```

```
end
```



F!

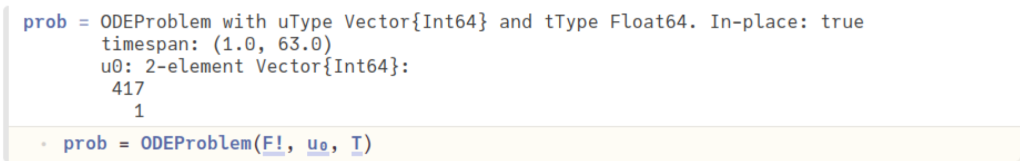
Правая часть нашей системы, p, t не используются. u[1] - S, u[2] - I

```
"Правая часть нашей системы, p, t не используются. u[1] - S, u[2] - I"
function F!(du, u, p, t)
    du[1] = - β * u[1] * u[2] / N + γ * u[2]
    du[2] = β * u[1] * u[2] / N - γ * u[2]
end
```

Рис. 3.9: Функция, задающая систему ОДУ

8. Описание модели SIS. Формирование проблемы (рис. 3.10).

```
prob = ODEProblem(F!, u⊠, T)
```



```
prob = ODEProblem with uType Vector{Int64} and tType Float64. In-place: true
       timespan: (1.0, 63.0)
       u0: 2-element Vector{Int64}:
        417
          1
prob = ODEProblem(F!, u₀, T)
```

Рис. 3.10: Формирование проблемы

9. Описание модели SIS. Решение проблемы (рис. 3.11).

```
sol = solve(prob, saveat=1)
```

sol =

	timestamp	value1	value2
1	1.0	417.0	1.0
2	2.0	416.851	1.14939
3	3.0	416.679	1.32094
4	4.0	416.482	1.5179
5	5.0	416.256	1.74396
6	6.0	415.997	2.00334
7	7.0	415.699	2.30084
8	8.0	415.358	2.64191
9	9.0	414.967	3.03275
10	10.0	414.52	3.48036
⋮ more			

```
sol = solve(prob, saveat=1)
```

Рис. 3.11: Решение проблемы

10. Формирование трех массивов: $S, I, time$ (рис. 3.12).

```
begin
    const ss = []
    const ii = []
    for u in sol.u
        s, i = u
        push!(ss, s)
        push!(ii, i)
    end
    time = sol.t
    time
end
```

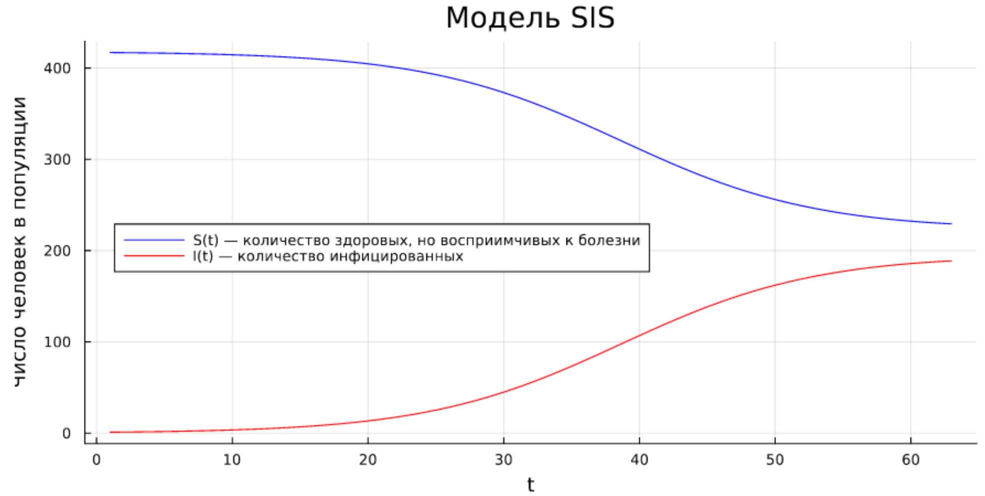
```
▸[1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.2, 4.4, 4
begin
    const ss = []
    const ii = []
    for u in sol.u
        s, i = u
        push!(ss, s)
        push!(ii, i)
    end
    time = sol.t
    time
end
```

Рис. 3.12: Формирование трех массивов: $S, I, time$

11. Отрисовка графика модели SIS (рис. 3.13).

```
begin
    fig3 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:left,
        plot_title="Модель SIS"
    )

    Plots.plot!(
        fig3[1],
        time,
        [ss, ii],
        color=[:blue :red],
        xlabel="t",
        ylabel="Число человек в популяции",
        label=["S(t) — количество здоровых, но восприимчивых к болезни" "I(t
```

```
    )
end
```



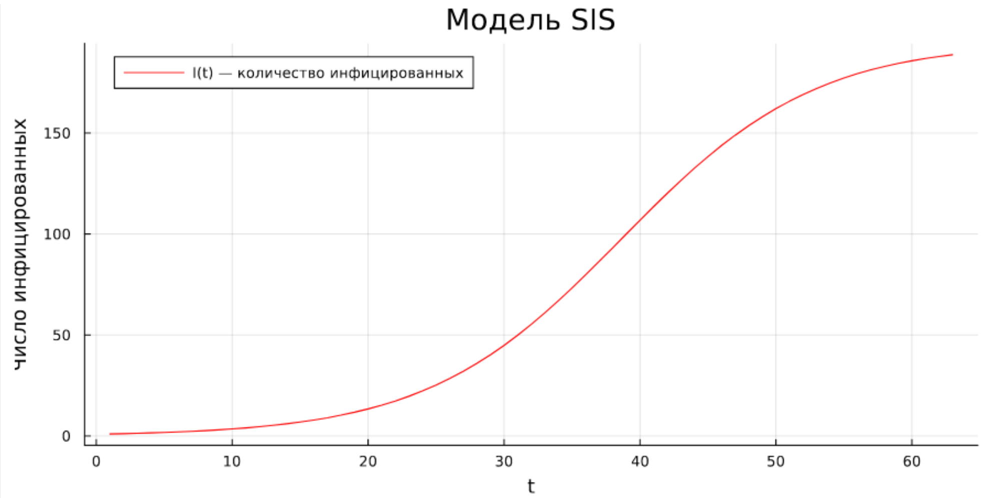

```
begin
    fig3 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:left,
        plot_title="Модель SIS"
    )

    Plots.plot!(
        fig3[1],
        time,
        [ss, ii],
        color=[:blue :red],
        xlabel="t",
        ylabel="Число человек в популяции",
        label=["S(t) — количество здоровых, но восприимчивых к болезни" "I(t) —
        количество инфицированных"]
    )
end
```

Рис. 3.13: Отрисовка графика модели SIS

12. Отрисовка графика модели SIS без учета $S(t)$ (рис. 3.14).

```
begin
    fig4 = Plots.plot(
        dpi=150,
        grid=:xy,
```

```
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Модель SIS"
    )

    Plots.plot!(
        fig4[1],
        time,
        ii,
        color=:red,
        xlabel="t",
        ylabel="Число инфицированных",
        label="I(t) — количество инфицированных"
    )
end
```



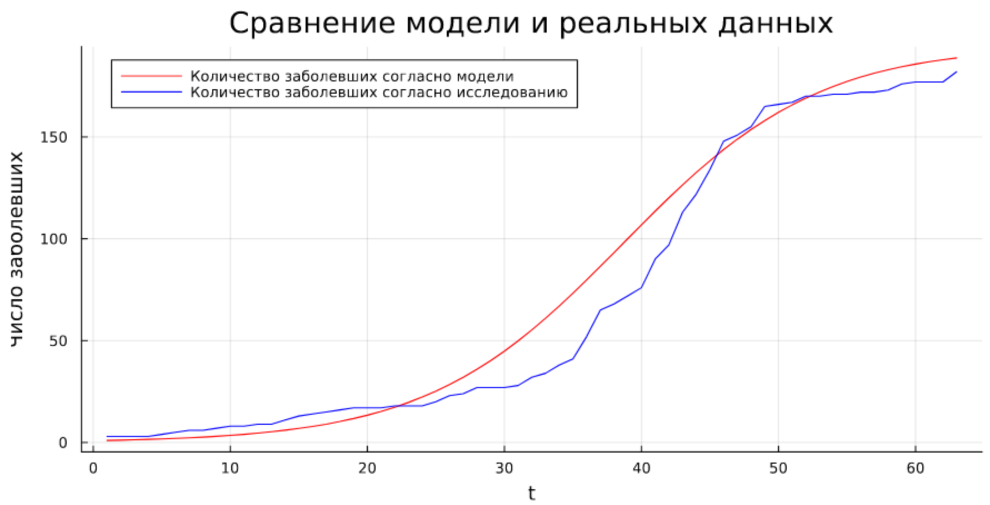

```
begin
    fig4 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Модель SIS"
    )

    Plots.plot!(
        fig4[1],
        time,
        ii,
        color=:red,
        xlabel="t",
        ylabel="Число инфицированных",
        label="I(t) — количество инфицированных"
    )
end
```

Рис. 3.14: Отрисовка графика модели SIS без учета $S(t)$

13. Отрисовка графика модели и графика реальных данных на одном графике. Как мы видим, результат моделирования крайне схож с найденными историческими данными из исследования распространения гриппа в одном из колледжей [9] (рис. 3.15).

```
begin
    fig5 = Plots.plot(
        dpi=150,
        grid=:xy,
```

```
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Сравнение модели и реальных данных"
    )

    Plots.plot!(
        fig5[1],
        days,
        [ii, df_sis],
        color=[:red :blue],
        xlabel="t",
        ylabel="Число заболевших",
        label=["Количество заболевших согласно модели" "Количество заболевши
    )
end
```




```
begin
    fig5 = Plots.plot(
        dpi=150,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:topleft,
        plot_title="Сравнение модели и реальных данных"
    )

    Plots.plot!(
        fig5[1],
        days,
        [ii, df_sis],
        color=[:red :blue],
        xlabel="t",
        ylabel="Число заболевших",
        label=["Количество заболевших согласно модели" "Количество заболевших
        согласно исследованию"]
    )
end
```

Рис. 3.15: Сравнение модели и реальных данных

14. Экспорт всех графиков в изображения (рис. 3.16).

```
begin
    Plots.savefig(fig1, "../fig1")
    Plots.savefig(fig2, "../fig2")
    Plots.savefig(fig3, "../fig3")
    Plots.savefig(fig4, "../fig4")
  Plots.savefig(fig5, "../fig5")
```

```
end
```

```
"/media/sf__/Доклад/presentation/fig5.png"
begin
    Plots.savefig(fig1, "../fig1")
    Plots.savefig(fig2, "../fig2")
    Plots.savefig(fig3, "../fig3")
    Plots.savefig(fig4, "../fig4")
    Plots.savefig(fig5, "../fig5")
end
```

Рис. 3.16: Экспорт всех графиков в изображения

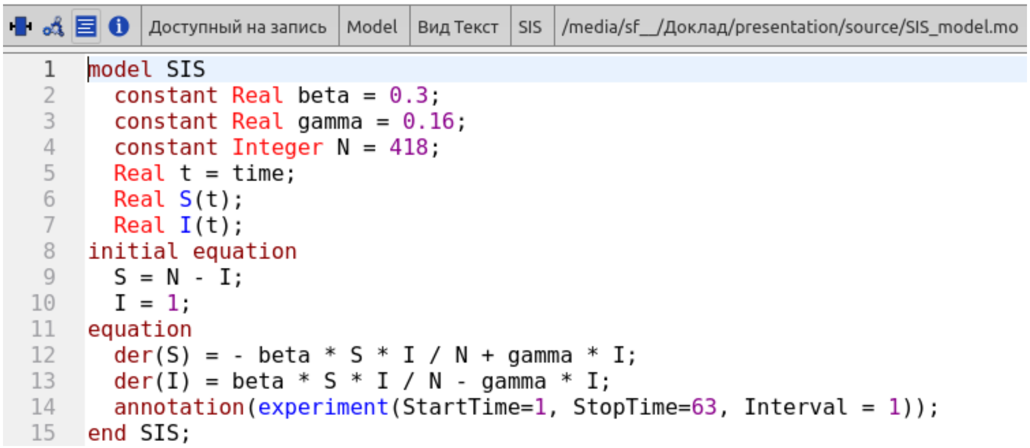
3.2 Modelica

По аналогии с Pluto пишем программу, воспроизводящую измененную модель SIR на языке моделирования Modelica с использованием ПО OpenModelica.

1. Код на языке моделирования Modelica: задаем название модели; определяем коэффициенты β и γ; численность популяции N; функции, зависящие от времени, $S(t)$ и $I(t)$; начальные условия; систему уравнений; начальное/конечное время и шаг симуляции (рис. 3.17).

```
model SIS
  constant Real beta = 0.3;
  constant Real gamma = 0.16;
  constant Integer N = 418;
  Real t = time;
  Real S(t);
  Real I(t);
initial equation
  S = N - I;
  I = 1;
equation
  der(S) = - beta * S * I / N + gamma * I;
```

```
    der(I) = beta * S * I / N - gamma * I;
    annotation(experiment(StartTime=1, StopTime=63, Interval = 1));
end SIS;
```




```
model SIS
  constant Real beta = 0.3;
  constant Real gamma = 0.16;
  constant Integer N = 418;
  Real t = time;
  Real S(t);
  Real I(t);
initial equation
  S = N - I;
  I = 1;
equation
  der(S) = - beta * S * I / N + gamma * I;
  der(I) = beta * S * I / N - gamma * I;
  annotation(experiment(StartTime=1, StopTime=63, Interval = 1));
end SIS;
```

Рис. 3.17: Код на языке моделирования Modelica

2. Лицезреем результат в виде двух графиков: зависимости S и I от времени и зависимости I от времени (рис. 3.18, 3.19).

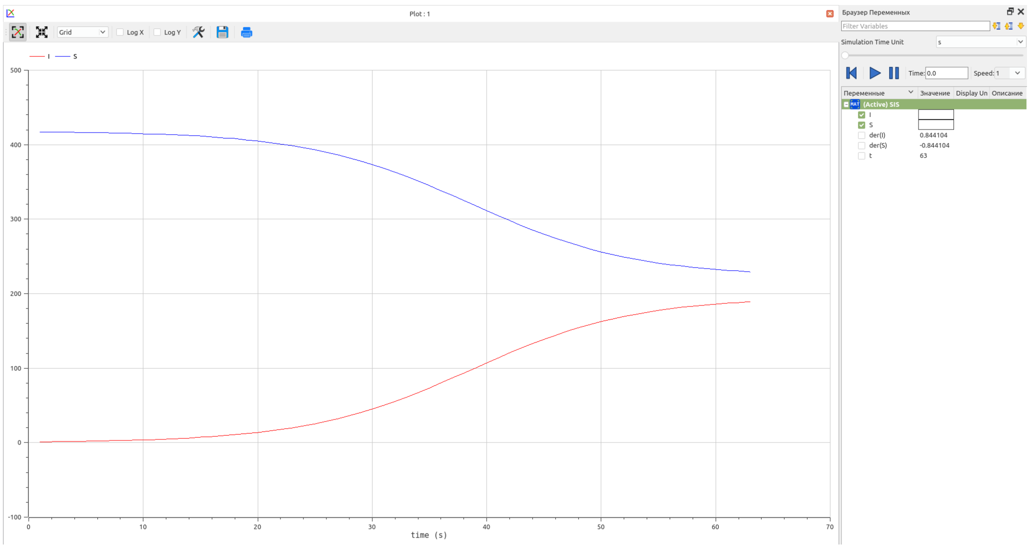


Рис. 3.18: Зависимость S и I от времени

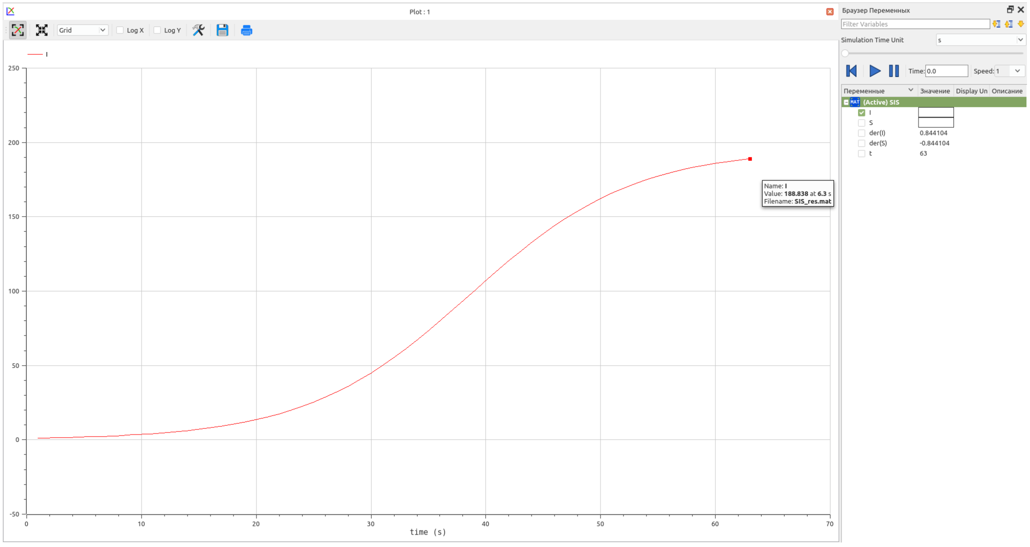


Рис. 3.19: Зависимость I от времени

4 Выводы

Изучил математическую модель заражения SIS. Используя функционал языка программирования Julia вместе с дополнительными библиотеками (DifferentialEquations, Plots), языка моделирования Modelica, а также интерактивного блокнота Pluto и программного обеспечением OpenModelica, описал модель заражения SIS. Сравнил описанную математическую модель с реальными данными о заражении.

Список литературы

1. Зараза, гостья наша [Электронный ресурс]. N + 1 Интернет-издание, 2019. URL: https://nplus1.ru/material/2019/12/26/epidemic-math.
2. Compartmental models in epidemiology [Электронный ресурс]. Wikimedia Foundation, Inc., 2023. URL: https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology.
3. An SIS model [Электронный ресурс]. Jeffrey M. Moehlis 2002-10-14. URL: https://sites.me.ucsb.edu/~moehlis/APC514/tutorials/tutorial_seasonal/node2.html.
4. ПРИМЕНЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ДЛЯ ПРОГНОЗИРОВАНИЯ ЭПИДЕМИОЛОГИЧЕСКОЙ СИТУАЦИИ ПО COVID-19 В РОСТОВСКОЙ ОБЛАСТИ [Электронный ресурс]. ФБУН «Ростовский научно-исследовательский институт микробиологии и паразитологии» Роспотребнадзора. URL: https://covid19.neicon.ru/files/3881.
5. Basic reproduction number [Электронный ресурс]. Wikimedia Foundation, Inc., 2023. URL: https://en.wikipedia.org/wiki/Basic_reproduction_number.
6. R0: How Scientists Quantify the Intensity of an Outbreak Like Coronavirus and Its Pandemic Potential [Электронный ресурс]. University of Michigan, 2020. URL: https://sph.umich.edu/pursuit/2020posts/how-scientists-quantify-outbreaks.html.
7. Estimates of the reproduction number for seasonal, pandemic, and zoonotic influenza: a systematic review of the literature [Электронный ресурс]. BioMed Central Ltd, 2014. URL: https://bmcinfectdis.biomedcentral.com/articles/10.1186/1471-2334-14-480.

8. Конструирование эпидемиологических моделей [Электронный ресурс]. Habr, 2021. URL: https://habr.com/ru/post/551682/.

9. Infectious disease in a total institution: a study of the influenza epidemic of 1978 on a college campus [Электронный ресурс]. Public Health Reports, 1982. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1424282/.