

感染症モデルの 微分方程式とシミュレーション

中央大学 理工学部物理学科四年 香取研究室

森澤良樹

はじめに

私は教授より渡された感染症数理モデルに関する資料を拝見し、微分方程式によって感染症の流行予想が行えるという部分に興味を惹かれた。

そこから、数理モデルの立式とそのシミュレーションについて研究することにした。以下を順を追って説明したいと思う。

- SIRモデルの導出
- SIRモデルのシミュレーション
- 免疫を獲得できないモデル(SISモデル)について

微分方程式の導出

まず、SIRモデルとはS(感受性のある人)、I(感染者)、R(隔離者)の三つの区分に分けて流行予想を行うものである。その推移は以下の微分方程式

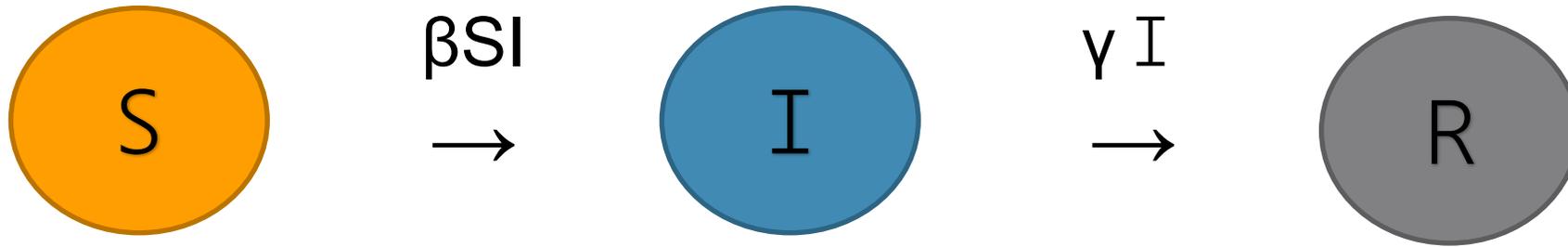


図1,SIRモデル

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \quad \frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \quad \frac{dR(t)}{dt} = \gamma I(t)$$

によって与えられる[1]が、これがどのようにして導出されているかを考える。

微分方程式の導出

最も単純なSIRモデルは以下のような条件を想定としている。

条件1：感染症以外での人口の変動はしない。

条件2：一度感染症から回復した場合、免疫を獲得して再感染しない。

これは非常に短期的な流行をする伝染病に対してよく使われるモデルである。

母集団を $S(t), I(t), R(t)$ に分けた時、それらの和は条件1より

$$S(t) + I(t) + R(t) = N(\text{定数}) \quad \dots(1)$$

と表すことができる[1]。

微分方程式の導出

続いてS(t)、I(t)、R(t)がどのように遷移するかを考える。感染者が増えることは数学的にはS(t)が減りその分I(t)が増えることになる。このS(t)の減少率は以下のような論理で数式に表すことができる。

- 感染性のある人が多いほど、感染者は増加しやすくなる。
→S(t)の減少率はI(t)に比例。
- 感受性のある人が多いほど、感染者は増加しやすくなる。
→S(t)の減少率はS(t)に比例。

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \quad \dots(2)$$

という方程式として表すことができる(βは比例定数)。感染のしやすさは年齢や環境、体質等によって変わるが、係数βはその平均を取った「感染率」とみなせる。

微分方程式の導出

同様に $R(t)$ についても、感染者が多いほど、免疫獲得(死亡)者も増えやすくなるために

$$\frac{dR(t)}{dt} = \gamma I(t) \quad \dots(3)$$

と表せる(γ は比例定数)。この γ も β と同様に感染者が感染性のなくなる平均確率であり、回復率や隔離率などと呼ばれる。

最後に $I(t)$ の変化率であるが、(1)を時間微分すると

$$\frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt} = 0$$

となり、後はこの式と(2)(3)より

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \quad \dots(4)$$

と $I(t)$ の時間変化を導ける。

微分方程式の導出

以上のように、微分方程式はどのパラメータに比例して変動するかという考えを元にできている。これを応用すれば感染症以外での人口の増減やワクチン接種などの影響に対しても議論ができる[1][2]。

表1, SIRの拡張例

区分	基本式	流入人口	流出人口	新生児ワクチン	ワクチン
$\frac{dS(t)}{dt} =$	$-\beta S(t)I(t)$	$+\epsilon N(t)$	$-\mu S(t)$	$-\nu\epsilon N(t)$	$-wS(t)$
$\frac{dI(t)}{dt} =$	$+\beta S(t)I(t) - \gamma I(t)$		$-\mu I(t)$		
$\frac{dR(t)}{dt} =$	$+\gamma I(t)$		$-\mu R(t)$	$+\nu\epsilon N(t)$	$+wS(t)$

SIRモデルの各区間の推移

S, I, Rがどういう推移をするのか、微分方程式から推察してみる。S、I、 β 、 γ 全ての値が非負であることから

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \quad \dots S \text{ は単調減少}$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \quad \dots I \text{ は } S = \gamma/\beta \text{ となるまでは増加、以後は減少を続ける}$$

$$\frac{dR(t)}{dt} = \gamma I(t) \quad \dots R \text{ は単調増加}$$

と、考えられる。

SIRモデルのシミュレーション

方程式からある程度の特徴を把握した上で、時間変化を描画してみたい。

SIRモデルのような微分方程式は、基本的に積分によって厳密解を求めることはできない[3]。しかし、厳密に解くことのできない数理モデルであっても、初期値を与えればその時間変化を求めることができる。

方法としては近似式を用いる。

$$f(a+t) = f(a) + tf'(a)$$

感染症数理モデルは各変数を微分した $f'(t)$ が既に分かっている状態なので、初期のパラメータ $f(0)$ を任意に設定すれば近似曲線が描ける。

今回、それをPythonのプログラムで行った。

SIRモデルのシミュレーション

```
1  from matplotlib import pyplot as plt
2  s = 0.99 #s(0)
3  i = 0.01 #i(0)
4  r = 0 #r(0)
5  be= 1.7 #感染率
6  gn= 0.8 #隔離率
7  ds =0 #s=s(n)の時のs(n+1)
8  di =0 #i=i(n)の時のi(n+1)
9  dr =0 #r=r(n)の時のr(n+1)
10 lists=[s] #sのリスト
11 listi=[i] #iのリスト
12 listr=[r] #rのリスト
13 listt=[0] #tのリスト
14 tt=0.04 #刻み幅
15
16 for t in range(1000):
17     listt.append(t)
18     ds = s+tt*(-be*s*i)
19     di = i+tt*(be*s*i-gn*i)
20     dr = r+tt*(gn*i)
21     lists.append(ds)
22     listi.append(di)
23     listr.append(dr)
24     s = ds
25     i = di
26     r = dr
27 plt.plot(listt,lists )
28 plt.plot(listt,listi );
29 plt.plot(listt,listr )
30 plt.title('SIRmodel')
31 plt.xlabel('Elapse time')
32 plt.ylabel('Population rate')
33 plt.show()
```

図2,SIRモデルのシミュレーションプログラム

SIRモデルのシミュレーション

図2で考えなくてはいけないのは各パラメータはどのような値にすべきかである。

まず β と γ はシミュレートする状況によって変更する定数である。

S、I、Rの初期値に関して、感染症の特性から感受性のある人たちの集団の中に僅かに初期感染者が現れるという状況を考えたい。

したがって $S(0) \doteq N$ 、 $I(0) = (\text{微小量})$ 、 $R(0) = 0$ と置くのが適切である。さらに $N=1$ と置くことで各区分が母集団に対する割合として考えることができ、都合がよい。そのため以後は $N=1$ として話を進める。

SIRモデルのシミュレーション

さらに考える必要があるのは、計算の際に使用する刻み幅 Δt である。

近似式 $f(a+\Delta t) = f(a) + \Delta t f'(a)$ は Δt の値が小さい時にのみ用いることができ、 Δt が大きいと実際の値との差も大きくなってしまふ。 Δt はどの程度小さければよいのか、プログラムを出力しながら確認していきたい。

確認方法は単純である。(4)式より、 $S = \gamma/\beta$ で I がピークに達するはずである。プロットした曲線がそうなっているかどうかで見ればよい。

また、 Δt の範囲rangeも刻み幅に合わせて変更しなくてはいけない。

SIRモデルのシミュレーション

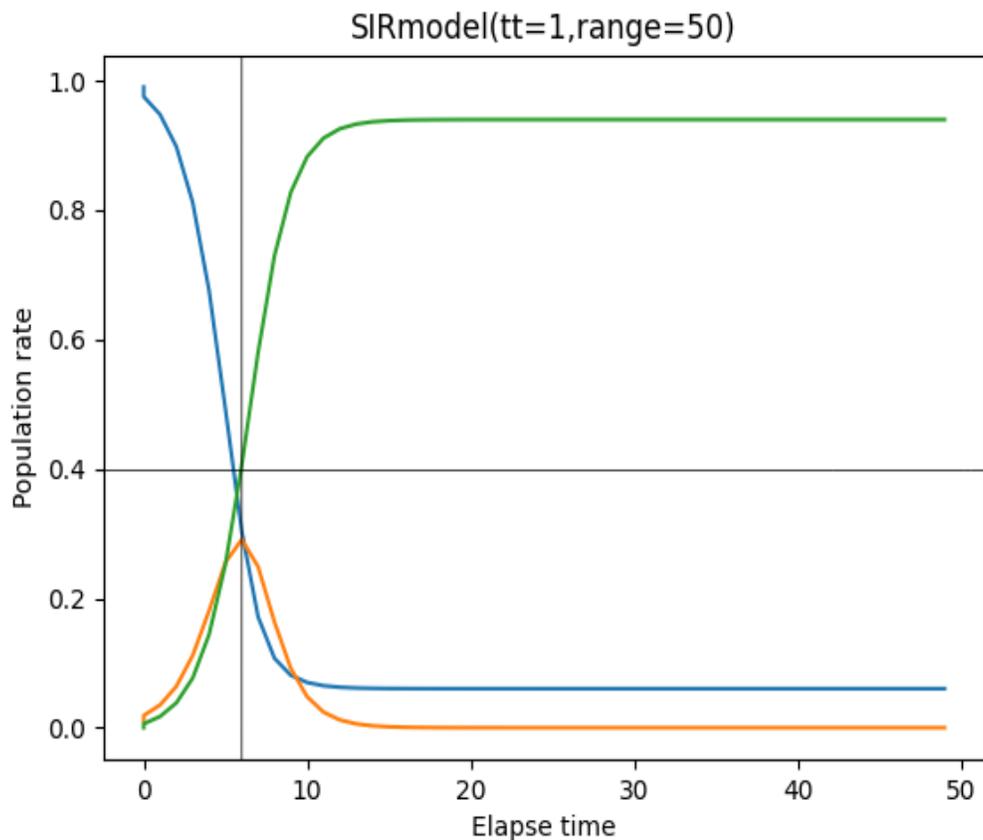


図3, $tt=1$ におけるSIRモデル

刻み幅が1の 때가 図3 である。なお $S(0)=0.99$ 、 $I(0)=0.01$ 、 $\beta=1.5$ 、 $\gamma=0.6$ である。理論通りなら I のピーク時に S が 0.4 でなくては いけない。

見るからに にガタガタしているほか、 $S=0.4$ と I のピーク値にズレが生じている。

中間発表時は Excel を使用し、この刻み幅でシミュレーションを行ったが、Excel では平滑な線になるよう補正してしまうので厳密ではないと分かった。

SIRモデルのシミュレーション

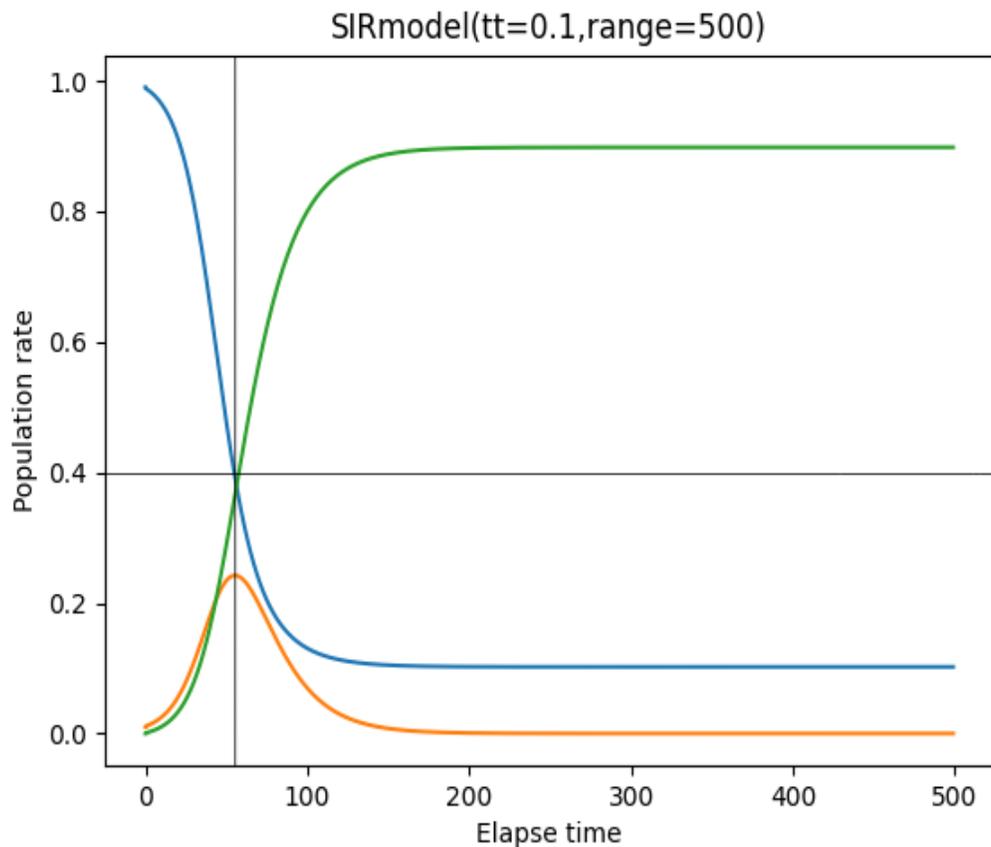


図4, $tt=0.1$ におけるSIRモデル

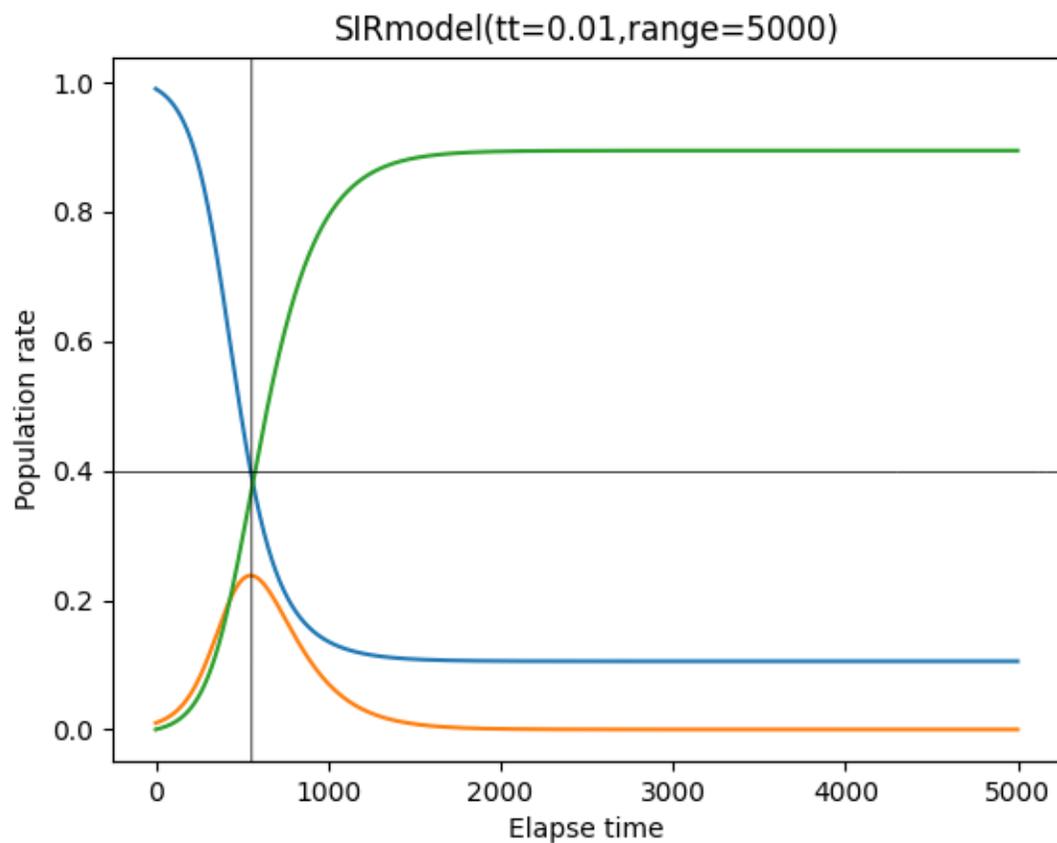
刻み幅が0.1の時が図4である。

線が滑らかになり、Sが0.4でIがピークに達しているので、ある程度正確な図であることが推察できる。

さらに0.01、0.001としたのが図5,図6である。図4と比べる若干変化が見られるが、図5と図6に差は(目測では)見られない。

そのため、 $tt=0.01$ であれば十分と判断し、以後の刻み幅は0.01で統一する。

SIRモデルのシミュレーション



図, $tt=0.01$ における SIR モデル

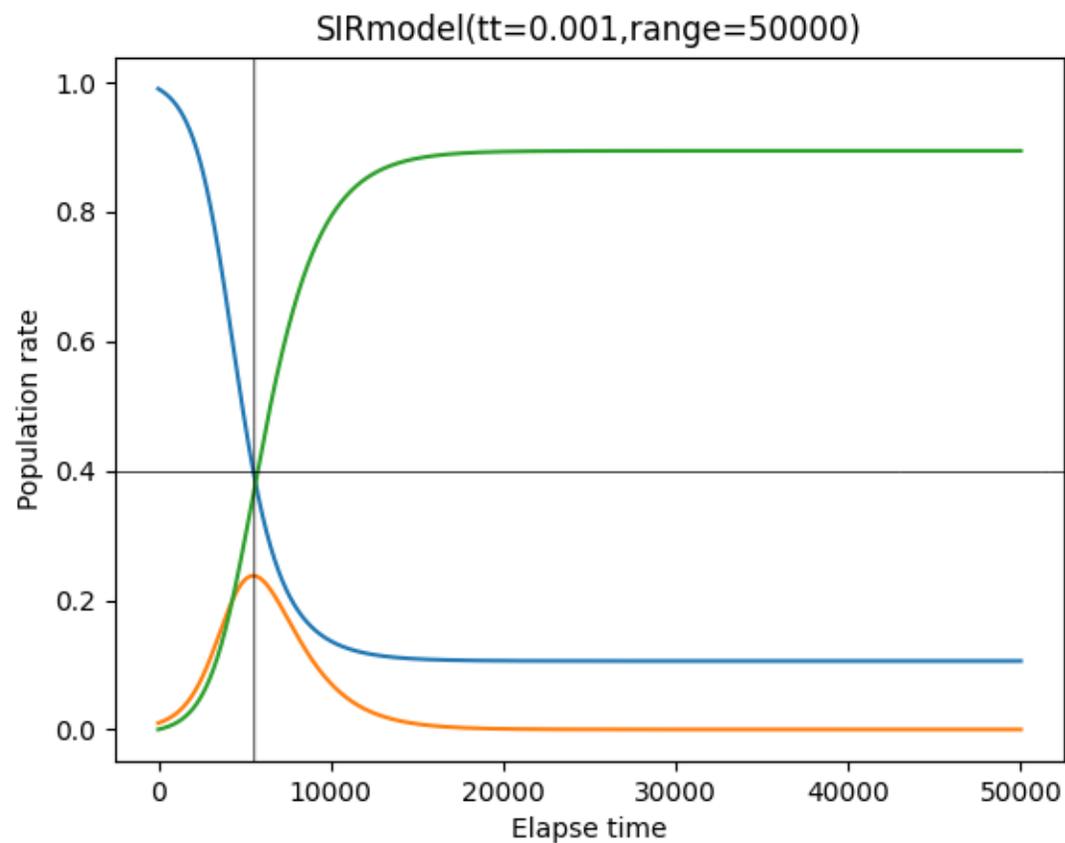


図6, $tt=0.001$ における SIR モデル

図からSIRモデルの特徴を読み取る

必要な刻み幅が分かったところで、 β や γ の値を変えてSIRモデルの特徴について考察する。

A($\beta=1.5, \gamma=0.6$)、**B**($\beta=1.5, \gamma=0.3$)、**C**($\beta=3.0, \gamma=0.6$)という3つのシミュレーションを行い、うち2つを比較する。それぞれ

Aと**B**... γ のみが異なっている→SIRモデルにおける γ の影響

Aと**C**... β のみが異なっている→SIRモデルにおける β の影響

Bと**C**... β, γ は異なっているが γ/β が同じ→SIRモデルにおける γ/β の影響

が調べられるはずである。

図からSIRモデルの特徴を読み取る

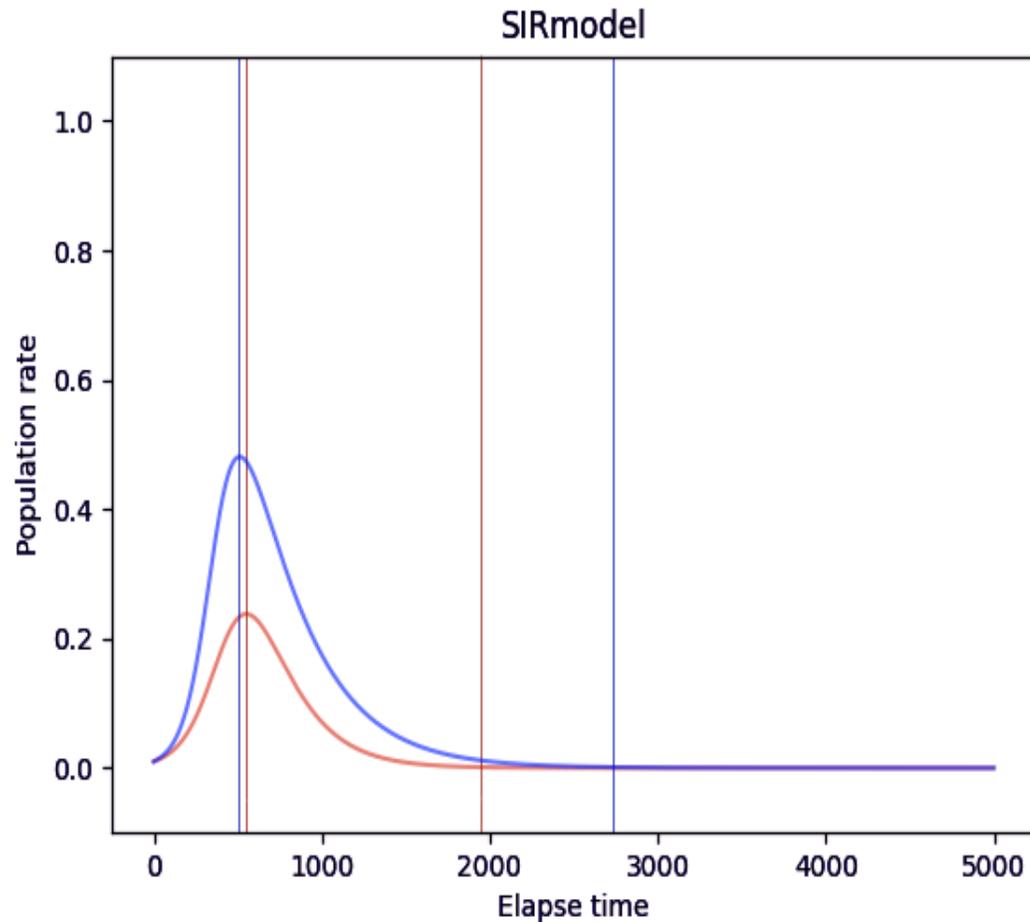


図7, モデルAとモデルBの比較

A($\beta=1.5, \gamma=0.6$)とB($\beta=1.5, \gamma=0.3$)の比較を行ったのが図7である。

Bのほうががピークに達するまでの時間が若干早く、逆に収束するまでの時間はかなり遅いことが分かる。

このことから、 γ は感染者のピークから収束するまでの時間に大きく関わる係数であると分かる。

図からSIRモデルの特徴を読み取る

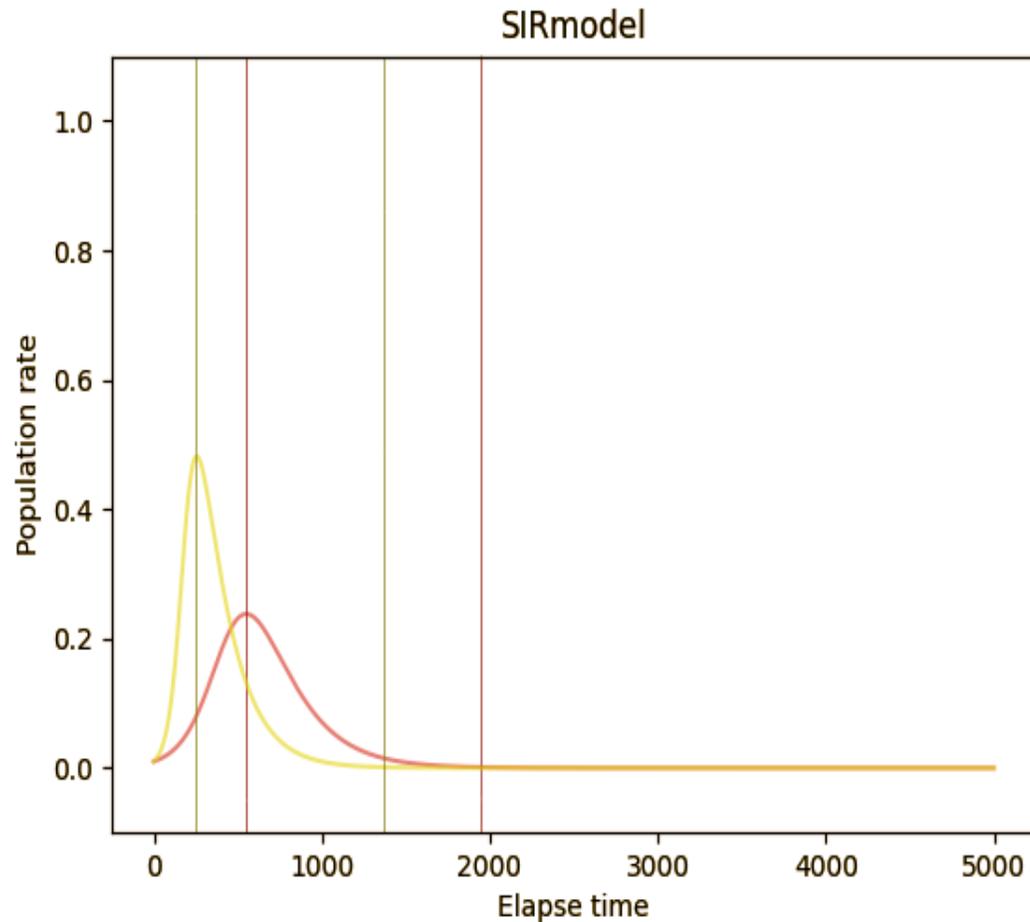


図8, モデルAとモデルCの比較

A($\beta=1.5, \gamma=0.6$)とC($\beta=3.0, \gamma=0.6$)の比較を行ったのが図8である。

Cのほうががピークに達するまでの時間、収束するまでの時間が共に早いことが分かる。

β は感染初期からピークに達するまでの時間、ピークから収束するまでの時間、両方に大きく関わる係数であると分かる。

図からSIRモデルの特徴を読み取る

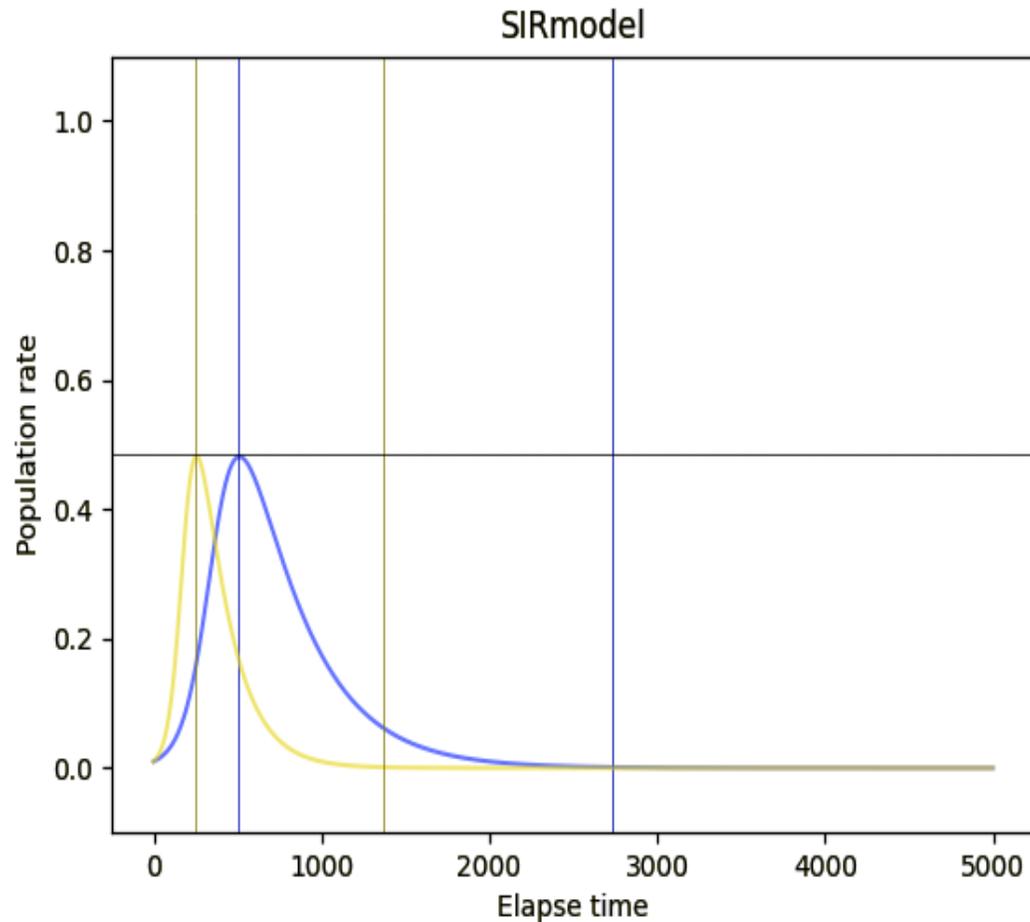


図9, モデルBとモデルCの比較(I)

$B(\beta=1.5, \gamma=0.3)$ と $C(\beta=3.0, \gamma=0.6)$ の比較を行ったのが図9である。

感染者のIのピークが来る時間に違いはあるものの、ピーク値はほぼ同値であることが分かる。

Iがピークの時、微分方程式よりSも γ/β で同値。さらに定義より $R=N-I-S$ なのでRも同値となる。

図からSIRモデルの特徴を読み取る

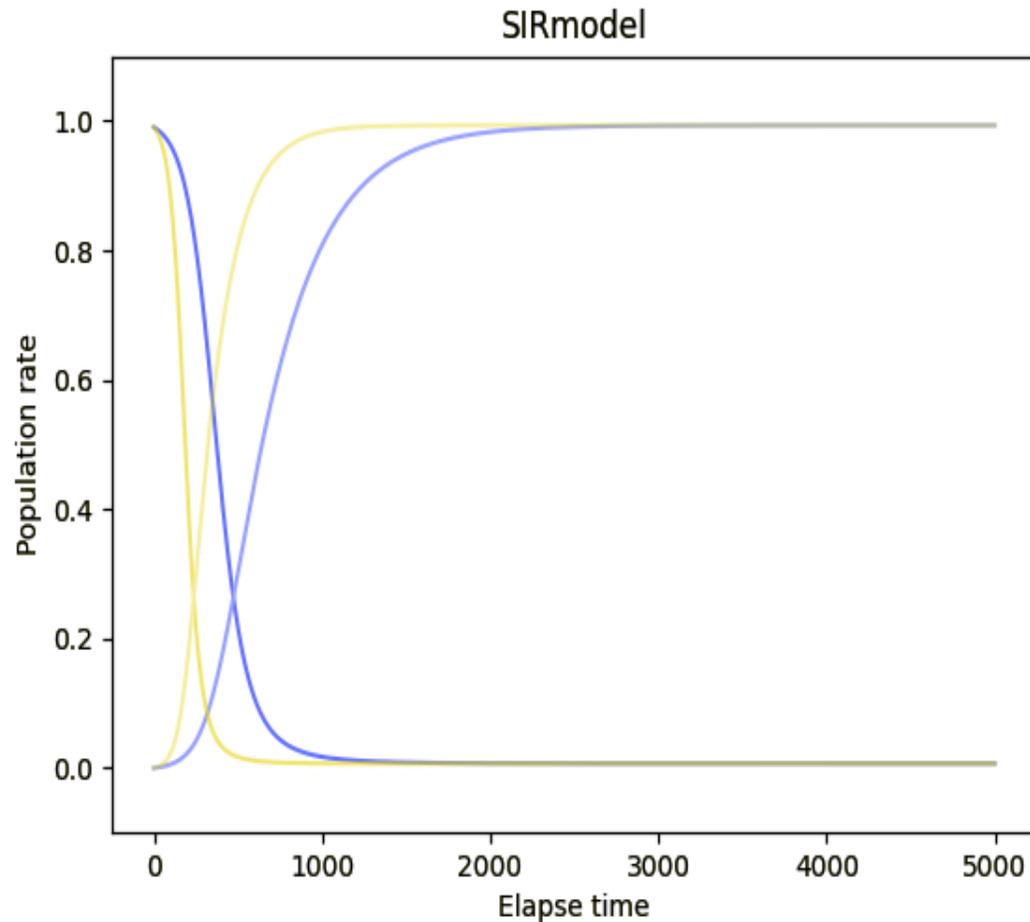


図10, モデルBとモデルCの比較(S,R)

さらに同モデルのSとRの比較も行って
みる。

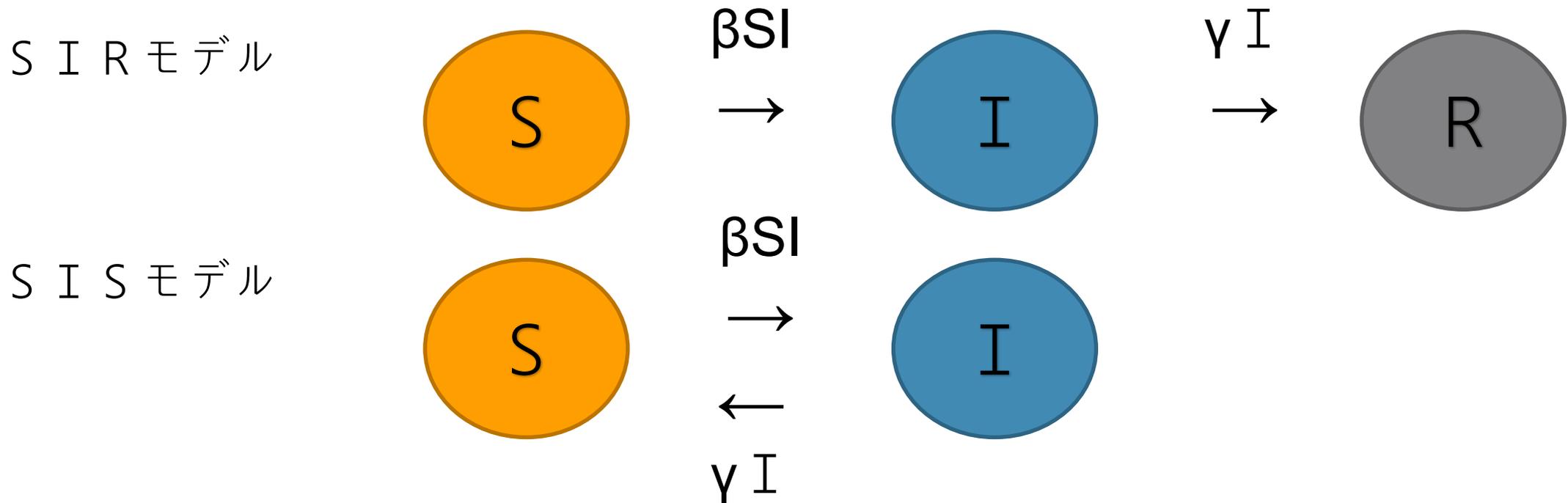
最終的なSとRの値、すなわち累計感染
者数が同じであった。(省略しているがA
との比較ではこうならなかった)

以上より、 γ/β にはIが最大時の各区分
の値、および累計感染者数に影響する係
数であると分かる。

SISモデルとは

SIRモデル以外にも感染症モデルはある。免疫獲得ができない感染症に対して用いる、SISモデルというものを紹介する[4]。

図11, SIRモデルとSISモデルの違い



SISモデルとは

免疫のできない感染症の例：ノロウイルス

厳密には免疫自体はできるが、半年程度で消えてしまう。加えてノロウイルスには沢山の型があり、免疫を持っていても別の型なので機能しない場合が多い。そのため、何度も感染する可能性がある[5][6]。

最初に発見されたのは**50**年以上前であるが、現在でも毎年数千人以上の感染者が確認されている[7]。

表2, ノロウイルスによる食中毒の患者数(参考)

年	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
患者数	10,874	13,904	8,619	17,632	12,672	10,506	14,876	11,397	8,496	8,475	6,889

SISモデルの微分方程式

図11より、SISモデルの微分方程式は

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) + \gamma I(t) = -(\beta S(t) - \gamma)I(t) \quad \dots(5)$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) = (\beta S(t) - \gamma)I(t) \quad \dots(6)$$

と、表すことができる[4]。

$S = \gamma/\beta$ で $S(t)$ も $I(t)$ も変化量が0になる。そのためSISモデルは最終的に新規感染者と回復者の数が釣り合う動的平衡状態となる。

SISモデルの微分方程式

さらにSISモデルの場合、微分方程式を解いて感染者数Iと時刻tの関係性を導くことができる[4]。(以下、 $S(t)=S, I(t)=I$ と表記する)

まず、SISモデルでは人口変動を考えていない、従って

$$S + I = N(\text{定数})$$

である。これより

$$S = N - I$$

と置ける。(6)式にこれを代入し、

$$\frac{dI}{dt} = \beta(N-I)I - \gamma I$$
$$\frac{dI}{(-\beta I + \beta N - \gamma)I} = dt$$

部分分数分解を用いて

$$\frac{1}{\beta N - \gamma} \left(\frac{1}{I} + \frac{\beta}{-\beta I + \beta N - \gamma} \right) dI = dt \quad \dots (7)$$

SISモデルの微分方程式

(7)式は変数分離法より、両辺をそれぞれの変数で積分すれば求めることができる。

$$\frac{1}{\beta N - \gamma} (\text{Log}|I| + \text{Log}|-\beta I + N\beta - \gamma|) + c = t \quad \dots (8)$$

最後に $I=I(0)$ で $t=0$ になるよう c を設定し、(8)式における I の範囲は初期値 $I(0)$ から $N-\gamma/\beta$ であることを考えれば、図示することができる。

これ以上の変形は難しいが、 t と I の関係を数式的に表すことができた。

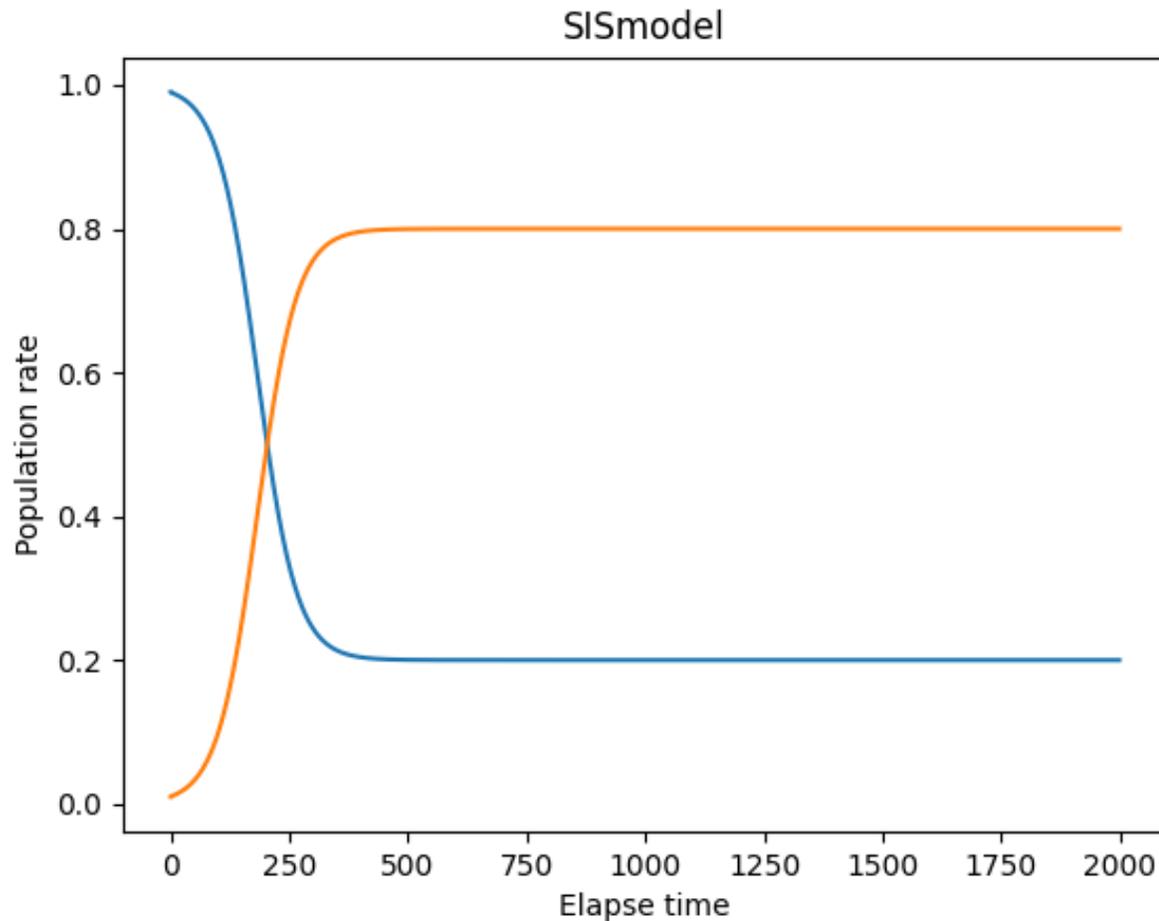
SISモデルのシミュレーション

SISモデルはSIRモデルと同様にシミュレーションをすることができる。

```
1  from matplotlib import pyplot as plt
2  s = 0.999 #s(0)
3  i = 0.001 #i(0)
4  be= 2.1 #感染率
5  gn= 0.5 #隔離率
6  ds =0 #s=s(n)の時のs(n+1)
7  di =0 #i=i(n)の時のi(n+1)
8  lists=[s] #sのリスト
9  listi=[i] #iのリスト
10 listt=[0] #tのリスト
11 tt=0.1 #刻み幅
12
13 for t in range(500):
14     listt.append(t)
15     ds = s+tt*(-be*s*i+gn*i)
16     di = i+tt*(be*s*i-gn*i)
17     lists.append(ds)
18     listi.append(di)
19     s = ds
20     i = di
21 plt.plot(listt,lists )
22 plt.plot(listt,listi );
23 plt.title('SISmodel')
24 plt.xlabel('Elapse time')
25 plt.ylabel('Population rate')
26 plt.show()
```

図12,SISモデルのシミュレーションプログラム

SISモデルのシミュレーション



シミュレーション結果が左図となる。今回は $\beta=3.0$ 、 $\gamma=0.6$ で行った。

前述の通り $S=\gamma/\beta=0.2$ で動的平衡状態となり、特に収束を見せる様子がない推移となる。

図13, SISモデルのシミュレーション ($S(0)=0.99, I(0)=0.01, \beta=3.0, \gamma=0.6$)

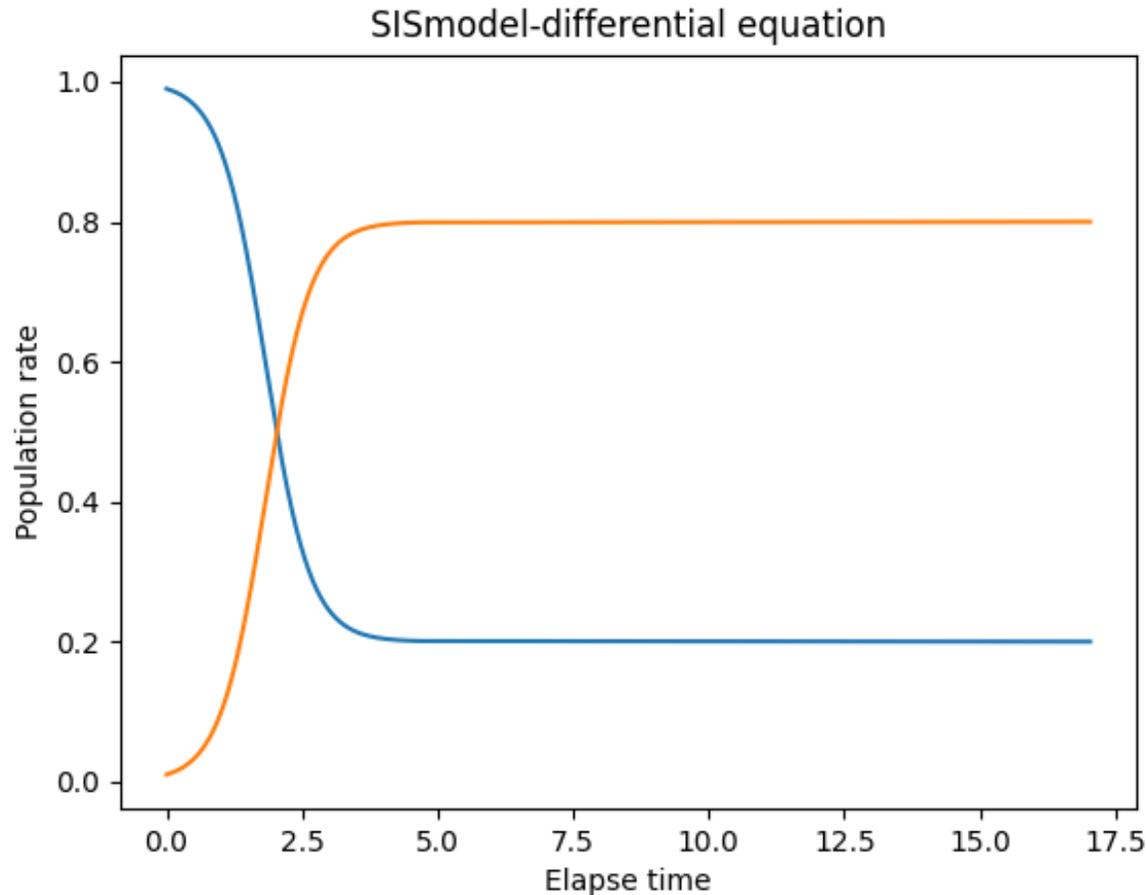
SISモデルのシミュレーション

```
sers > yoshiki > python_lesson > SIS-EQ.py
import matplotlib.pyplot as plt
import math
import numpy as np
be= 3.0 #感染率
gn= 0.6 #隔離率
Io= 0.01 #初期感染者数
c= (1/(be-gn))*(np.log(abs(Io))-np.log(abs(Io-(be-gn)/be)))
i = np.linspace( 0.01, 1-gn/be, 1000) # linspace(mi, ma, N) miからmaをN分割
t =(1/(be-gn))*(np.log(abs(i))-np.log(abs(i-(be-gn)/be)))-c
plt.plot(t,1-i)
plt.plot(t,i)
plt.title('SISmodel-differential equation')
plt.xlabel('Elapse time')
plt.ylabel('Population rate')
plt.show()
import numpy as np
```

図12とは別に、(8)式を用いたSISモデル推移の確認を試みる。

図14, i と t の関係式を図示するプログラム

SISモデルのシミュレーション



結果が左図となる。
グラフの形は変わらないが、図13と比較すると、動的平衡状態になるまでの時間に違いがある。これは時間軸の単位について考慮されていないためである。

図15, SISモデルのシミュレーション ($S(0)=0.99, I(0)=0.01, \beta=3.0, \gamma=0.6$)

最後に

9月頃から感染症数理モデルにテーマを決め、そこから微分方程式を中心に**SIS**モデルやシミュレーションプログラムについて研究することができた。

ただ、興味を持ったことを優先したために明確なゴールがないまま研究を進めたのは反省点である。

コロナ禍で思うようにいかないことが多々あったが、自主的にテーマや課題を決めて研究を行えたのは非常に良い経験ができたと思う。

参考文献

- [1] 稲葉寿 微分方程式と感染症数理疫学 数理理科学 NO. 538, APRIL 2008
(https://www.ms.u-tokyo.ac.jp/~inaba/inaba_science_2008.pdf)
- [2] 石川昌明 時間遅れを考慮した確率SIRモデルの安定性解析 システム成語情報学会論文誌, Vol.30, No.4, pp.115-121 2017
(https://www.jstage.jst.go.jp/article/iscie/30/4/30_115/_pdf)
- [3] 数理モデルとシミュレーション 小川知之 宮路智行 2020/12/25初版発行
- [4] 佐々木徹 拡散効果のある SIS 感染症モデルの局所対策について 数理解析研究所講究録 1372 巻 2004 年 123-127
(<http://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/1372-19.pdf>)

参考文献

[5] ノロウイルスによる食中毒はなぜ無くならないのか 愛知医科大学 医学部 公衆衛生学講座 西尾治

(<http://www.mac.or.jp/mail/121001/01.shtml>)

[6] ノロウイルス感染症の拡大を防ぐには 国立感染症研究所感染疫学センター 第六室長 木村博一 情報誌けあ・ふるVOL.90 (2017/1)掲載

(<https://www.paramount.co.jp/learn/topics/detail/1>)

[7] ノロウイルスに関するQ & A 厚生労働省 平成30年5月31日

(https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryoushokuhin/syokuchu/kanren/yobou/040204-1.html)

[8] 西浦博・稲葉寿 感染症流行の予測：感染症数理モデルにおける定量的課題 統計数理研究所 統計数理(2006)第54巻第2号 461-480 改定2006年2月6日

(<https://www.ism.ac.jp/editsec/toukei/pdf/54-2-461.pdf>)

[9] 長田直樹 区画に無症状感染者を持つ感染症の数理モデル 東京女子大学 2020年7月23日 Version 1.4

(https://www.lab.twcu.ac.jp/~osada/math_model/math_model.pdf)