

承 诺 书

我们队完全明白，在完成期末考核作业中，队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括教师）研究、讨论与作业题目有关的问题。

我们知道，抄袭别人的成果是违反诚信规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守结业作业的基本规则，以保证大作业评阅的公正、公平性。如有违反诚信规则的行为，我们明白将会受到严肃处理。

我队的编号： MHSXJM-3-07

队员姓名/ 学号/工作占比：

1. 姓名： 张镕麒 学号： 2019081301026 工作占比： 47.7%
2. 姓名： 赵润桐 学号： 2019081301027 工作占比： 18.2%
3. 姓名： 张泽儒 学号： 2019080601029 工作占比： 34.1%

任课教师： 李良

基于 SEIR 模型的新冠疫情预测

摘要

新型冠状病毒肺炎（Corona Virus Disease 2019, COVID-19），简称“新冠肺炎”，世界卫生组织命名为“2019 冠状病毒病”，是指 2019 新型冠状病毒感染导致的肺炎。新冠疫情的爆发对政府及相关管理部门现有的管控方法和措施都提出了新的严峻挑战。如今半年已经过去，经济停止发展，人群被迫隔离。人们最关心的疫情的发展何时出现拐点？以及疫情何时能过去？我们的生活何时能恢复正常？

针对新冠疫情的传染过程，我们对其进行抽象概括，得出了一系列的传染病微分方程模型，分别是 SIR、SEIR、SEIRS 和改进 SEIR 模型。我们主要选取了 SIR 对其进行分析，得出了几点有实际意义的结论。

同时针对传染病学中的一个十分重要的量——实际传染数建立数学模型。通过前人对各种防疫措施防疫效果的研究，建立了实际传染数与防疫措施之间的线性关系。再通过分析各个国家疫情措施，得出对应的疫情发展趋势。

最后我们对模型进行运用、预测。首先对中国的疫情数据进行 Logistics 曲线拟合，得到近似的拐点，并根据对应时间分析对应防疫措施，进一步验证实际传染数模型的正确性。其次，应用先前提出的传染病模型，用差商代替微分，用蒙特卡洛最小二乘法求出对应微分方程组中的最优参数。从而求得其他国家的疫情拐点出现时间段。

关键词：SEIR 模型 蒙特卡罗法 数值计算 数据拟合 预测

目录

1 问题重述.....	1
2 问题分析.....	1
3 模型假设.....	1
4 名词解释和符号说明.....	2
5 模型建立与求解.....	3
5.1 传染系统的模型建立.....	3
5.1.1 由易感者、传染者和痊愈者构成的 SIR 模型.....	3
5.1.2 额外考虑了潜伏者的 SEIR 模型.....	6
5.1.3 额外考虑了痊愈者失去免疫力的 SEIRS 模型.....	6
5.1.4 额外考虑了隔离易感者、隔离潜伏者的 SEIR 模型.....	7
5.2 传染数的模型建立.....	8
5.2.1 基本传染数的影响因素.....	9
5.2.2 实际传染数的影响因素.....	9
5.2.4 考虑了隔离易感者、隔离潜伏者的 SEIR 模型的分析.....	11
6 模型检验与比较.....	12
6.1 中国疫情分析.....	12
6.1.1 中国疫情拐点出现时间判断.....	12
6.1.2 中国疫情拐点成因判断.....	12
6.2 美国疫情分析.....	12
6.2.1 美国疫情拐点成因判断.....	12
6.2.2 美国以轻拐点出现时间预测.....	13
7 模型评价.....	15
7.1 模型的优点.....	15
7.2 模型的缺点及改进方向.....	15

1 问题重述

COVID-19 对全球的影响深远，它是自 2003 年 SARS 病毒大流行以来，带来最严重的公共卫生威胁的一种呼吸道病毒。新冠疫情的爆发对政府及相关管理部门现有的管控方法和措施都提出了新的严峻挑战。¹人们最关心的疫情的发展何时出现拐点？以及疫情何时能过去？我们的生活何时能恢复正常？本次数学建模期末考核要求收集相关数据，建立有关疫情的相关模型，选取一个控制疫情较好的国家，分析判断其疫情出现拐点的时间及成因。并另外选择一个疫情并未明显出现拐点的国家，预测该国疫情拐点到达的时间。

2 问题分析

首先根据新冠疫情的特性，我们逐步深入地建立了传染系统中各类人群相互作用导致的数量变化的人群数量模型——SIR、SEIR、SEIRS。并分析了几种应对疫情的措施和环境因素对实际传染数的影响，分别建立了有关实际传染数的模型。结合以上模型——

针对问题一，我们选择了防控疫情工作进入尾声的中国，收集了中国新冠疫情相关的病例数据。由于累计确诊病例数随日期单调递增，所以对该数量进行函数拟合，再对该函数求拐点即可求出近似的疫情拐点。

针对问题二，我们根据实际情况，近似求解上述模型对应的参数以获得中国国内新冠病毒的实际传染数。再结合中国政府对疫情采取的各项措施和先前建立的实际传染数的模型，分析拐点出现的成因，以及各项举措的疫情控制的贡献。

针对问题三，我们选择了疫情尚未明显出现拐点的美国。根据美国政府可能采取的不同措施和民众对措施的不同遵守情况，我们考虑了两种可能的实际情况，并根据模型分别预测了美国疫情出现拐点的时间段。

3 模型假设

1. 因病毒死亡的病人与痊愈者不再次参与传染过程，可将死亡患者视作痊愈者。

2. 除因病毒导致的死亡外，不考虑自然情况下的生死与迁移，可认为在一段时间内一定区域的人口总数不变。

4 名词解释和符号说明

易感者 (Susceptible)：在传染系统中可被传染者感染的人群总称，简称为健康者。

感染者 (Exposed)：易感者被感染后一段时间内仍未拥有感染易感者的能力的人群总称。

传染者 (Infectious)：在传染系统中可传染的易感者的总称，又称为病人。

痊愈者 (Recovered)：传染者经过治疗并痊愈的人群总称，不再会被传染。

有效接触：指传染者接触易感者，使易感者变成感染者或传染者的过程。

免疫力丧失：指痊愈者经过一段时间后变成易感者的过程。

感染周期：平均意义下传染者成为痊愈者的时间。

基本传染数：在没有措施的情况下，一个病人在感染周期内能够传染的平均人数。

实际传染数：在人为干预的情况下，一个病人在感染周期内能够传染的平均人数。

表 1 符号说明

符号定义	符号说明
N	总人数
$s(t)$	时刻 t 易感者在总人数 N 中占的比例
$i(t)$	时刻 t 传染者在总人数 N 中占的比例
$r(t)$	时刻 t 痊愈者在总人数 N 中占的比例
$e(t)$	时刻 t 痊愈者在总人数 N 中占的比例
ω_1	潜伏者每天转变为感染者的比例
ω_2	痊愈者每天转变为易感者的比例
μ	日治愈率
λ	日接触率
σ	传染期接触数/基本传染数
S	易感者人数
E	潜伏者人数
I	传染者人数
R	痊愈者人数
S_q	隔离易感者数目
E_q	隔离潜伏者数目
H	住院患者数目/隔离感染者数目
R	康复人数
η	潜伏者向感染者的转化速率
k	隔离解除速率
c	接触率

q	隔离比率
β	感染率
δ_I	感染者的隔离速率
δ_q	隔离潜伏者向隔离感染者的转化速率
γ_I	感染者的恢复率
γ_H	隔离感染者的恢复速率
α	病死率

5 模型建立与求解

5.1 传染系统的模型建立

由于不同传染病的传染过程不同，我们针对新冠病毒的传染过程特征，逐步抽象提取出了 SEIR 模型，并以 SEIR 模型为基础，提出了额外考虑痊愈者失去免疫能力的 SEIRS 模型和额外考虑了对不同人群采取隔离措施的改进 SEIR 模型，通过求解各个模型分析出了不同参数下病毒的传染过程及结果。

5.1.1 由易感者、传染者和痊愈者构成的 SIR 模型

大多数传染病如天花、流感肝炎、麻疹等治愈后均有很强的免疫力，所以病愈的人既非健康者(易感染者)，也非病人。针对于新冠疫情而言，我们假设病人痊愈后会具有较强的免疫力，即不再具有易感性，已经退出传染系统。

人群分为易感者、传染者和痊愈者三类。时刻 t 三类人在总人数 N 中占的比例分别记作 $s(t)$ ， $i(t)$ 和 $r(t)$ 。病人的日有效接触率为 λ ，日治愈率为 μ ，传染期接触数²为 $\sigma = \frac{\lambda}{\mu}$ 。

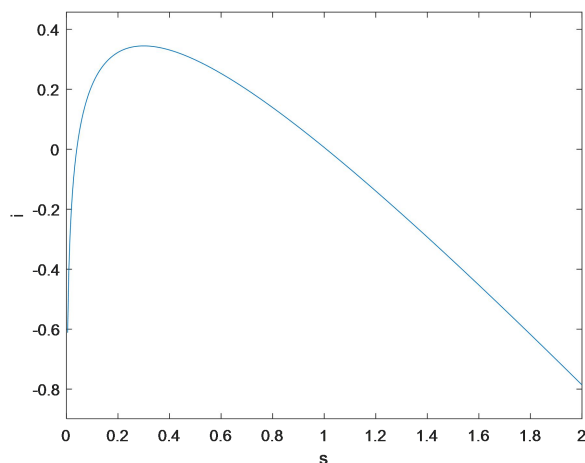
显然有，

$$s(t) + i(t) + r(t) = 1$$

人群间的转移过程如下微分方程所示：

$$\begin{cases} \frac{ds}{dt} = -\lambda si \\ \frac{di}{dt} = \lambda si - \mu i \\ \frac{dr}{dt} = \mu i \end{cases} \quad (1)$$

$i-s$ 的图形称为相轨线，如图所示：

图 1 $i-s$ 相轨线

随着时间进行, (s, i) 沿轨线自右向左运动。对 (1) 式进行整理, 消 t 有

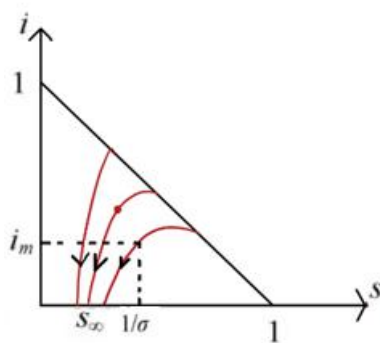
$$\frac{di}{dt} = \frac{\mu}{\lambda} s - 1 \quad i(s_0) = i$$

解得

$$i = \frac{\mu}{\lambda} \ln s - s + c \quad (\text{其中 } c = i_0 + s_0 - \frac{\mu}{\lambda} \ln s_0)$$

代入初值 $\lambda = 1, \mu = 0.3, i(0) = 0.02, s(0) = 0.98$, 得

$$\begin{cases} c = 1.0061 \\ i = 0.3 \ln s - s + 1.0061 \end{cases}$$

图 2 $i-s$ 相平面

相轨线在相平面上的定义域

$$D = \{(s, i) | s \geq 0, i \geq 0, s + i \leq 1\} \quad (2)$$

令 $\sigma = \frac{\mu}{\lambda}$, 则方程 (1) 可化为

$$\frac{di}{ds} = \frac{1}{\sigma} - 1$$

解得

$$i = s_0 + i_0 - s + \ln \frac{\left(\frac{s}{s_0} \right)}{\sigma} \quad (3)$$

在定义域 D 内, (3) 式表示的曲线即为相轨线, 其中箭头表示了随着时间 t 的增加 $s(t)$ 和 $i(t)$ 的变化趋向。

记 $s(t)$, $i(t)$ 和 $r(t)$ 在 $t \rightarrow \infty$ 时的极限值分别记作 s_∞ , i_∞ 和 r_∞ 。有如下结论成立。

1. 不论初始条件 i_0, s_0 如何, 病人终将消失, 即

$$i_\infty = 0 \quad (5)$$

从图形上看, 不论相轨线从 P1 或从 P2 点出发, 当 t 充分大时, 它都会与 s 轴相交。

2. 最终未被感染的健康者的比例为 s_∞ 。在 (3) 式中令 $i = 0$ 得, s_∞ 是方程

$$s_0 + i_0 - s_\infty + \frac{1}{\sigma} \ln \frac{s_\infty}{s_0} = 0 \quad (5)$$

在 $\left(0, \frac{1}{\sigma}\right)$ 内的根。在图形上 s_∞ 是相轨线与 s 轴在 $\left(0, \frac{1}{\sigma}\right)$ 内交点的横坐标。

3. 若 $\begin{cases} s_0 > \frac{1}{\sigma} & \text{则 } i(t) \text{ 先增加} \\ s_0 = \frac{1}{\sigma} & \text{则 } i(t) \text{ 达到最大值} \end{cases}$

$$i_m = s_0 + i_0 - \frac{1}{\sigma} (1 + \ln(\sigma s_\infty)) \quad (6)$$

当 $s > \frac{1}{\sigma}$ 时, $i(t)$ 减小且趋于 0, $s(t)$ 则一直单调减小至 s_∞ 。

4. 若 $s_0 < \frac{1}{\sigma}$, 则 $i(t)$ 单调减小至零, $s(t)$ 单调减小至 s_∞ 。如果仅考虑病人比例 $i(t)$ 有一段增长的时期视为传染病在蔓延, 那么 1 是一个阈值。即通过群体免疫使初始时刻的移出者比例 (即免疫者比例) r_0 ³, 就可以制止传染病的蔓延。当 $s_0 > \frac{1}{\sigma}$ (即 $\sigma > \frac{1}{s_0}$) 时, 传染病将蔓延。若减小传染期接触数 σ , 即提高阈值 $\frac{1}{\sigma}$, 使得 $s_0 < \frac{1}{\sigma}$ (即 $\sigma \leq \frac{1}{s_0}$), 传染病就不会蔓延。即使 $s_0 > \frac{1}{\sigma}$, 从 (5)、(6) 式可以看出, σ 减小时, s_∞ 增加, i_m 降低, 也控制了蔓延的程度。

在 $\sigma = \frac{\mu}{\lambda}$ 中: 人们的卫生水平越高, 日接触率入越小; 医疗水平越高, 日治愈率 μ 越大, 最终 σ 越小。所以提高卫生水平和医疗水平有助于控制传染病的

蔓延。

另一方面，定义 $\sigma s = \lambda s * \frac{1}{\mu}$ 为交换数，其含义是一个病人被 σs 个健康者交换。所以当 $s_0 \leq \frac{1}{\sigma}$ ，即 $\sigma s_0 \leq 1$ 时，必有 $\sigma s \leq 1$ ，既然交换数不超过 1，病人比例 $i(t)$ 不会增加，传染病不会蔓延。

5.1.2 额外考虑了潜伏者的 SEIR 模型

现实生活中，有些疾病易感者患病时并不会立刻爆发，而是先进入潜伏期，等潜伏期结束后，易感者才真正变为感染者。所以，对于这种疾病，就需要在 SIR 模型的基础上增加 $e(t)$ 表示潜伏者占总人数的比例。

人群分为易感者、潜伏者、传染者、痊愈者四类。时刻 t 四类人在总人数 N 中占的比例分别记作 $s(t)$ ， $e(t)$ ， $i(t)$ 和 $r(t)$ 。病人的日接触率为 λ ，日治愈率为 μ ，基本传染数为 $R_0 = \frac{\lambda}{\mu}$ 。潜伏者每天转变为感染者的比例为 w 。

综上所述，可得出如下方程组

$$\begin{cases} N \frac{ds}{dt} = -\lambda N i s \\ N \frac{de}{dt} = \lambda N i s - N w e \\ N \frac{di}{dt} = N w e - \mu N i \\ N \frac{dr}{dt} = \mu N i \end{cases}$$

分析过程与 SIR 模型类似。

5.1.3 额外考虑了痊愈者失去免疫力的 SEIRS 模型

大多数传染病在患者痊愈后都会具有很强的免疫力，但并非所有传染病都是终身免疫的，部分传染病的痊愈者依然具有在将来再次患病的可能，在 seirs 模型中，我们假设痊愈者在一定时间内具有很强的免疫力（不会患病），一段时间后，免疫力消失，可能再次患病。

人群分为易感者、潜伏者、传染者、痊愈者四类⁴。时刻 t 四类人在总人数 N 中占的比例分别记作 $s(t)$ ， $e(t)$ ， $i(t)$ 和 $r(t)$ 。病人的日接触率为 λ ，日治愈率为 μ ，基本传染数为 $R_0 = \frac{\lambda}{\mu}$ 。痊愈者每天重新转变为易感者的人数占痊愈者人数的

的比例为 w_2 潜伏者每天转变为感染者的比例为 w_1

综上所述，可得出如下方程组

$$\begin{cases} N \frac{ds}{dt} = -\lambda Nis + Nr w_2 \\ N \frac{de}{dt} = \lambda Nis - Nw_1 e \\ N \frac{di}{dt} = Nw_1 e - \mu Ni \\ N \frac{dr}{dt} = \mu Ni - Nr w_2 \end{cases}$$

分析过程与 SIR 模型类似。

5.1.4 额外考虑了隔离易感者、隔离潜伏者的 SEIR 模型⁵

人群分为易感者、隔离易感者、隔离潜伏者、潜伏者、感染者、住院患者、痊愈者七类，七类人数分别记作 S, S_q, E, E_q, I, H, R 。隔离解除速率为 k ，取 $k=1/14$ （考虑到隔离时间为 14 天），潜伏者到感染者的速率 η ，取 $\eta=1/7$ （考虑到潜伏到发病周期为 7 天）。 q 为隔离概率， β 为传染概率， c 为接触率。模型示意图

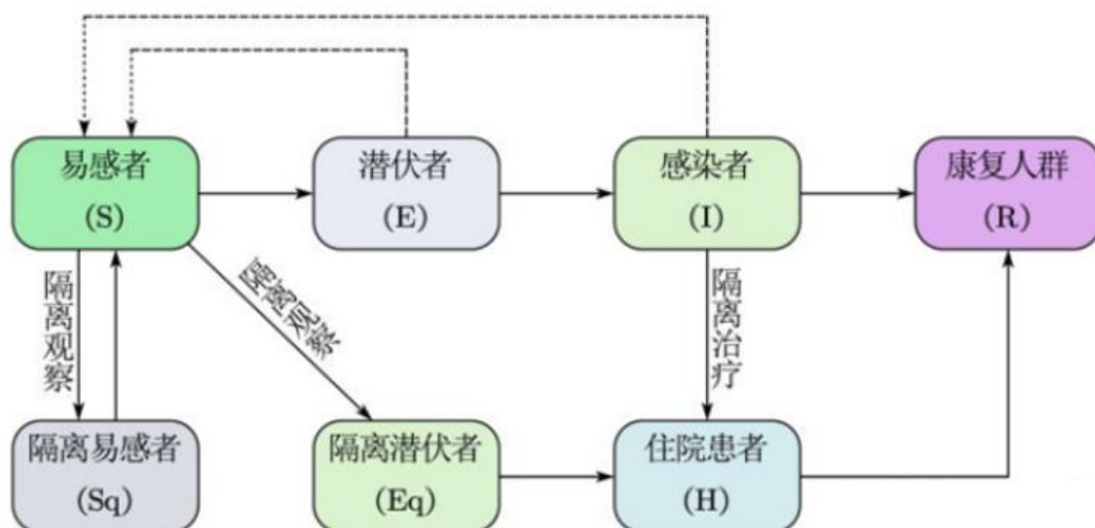


图 3 改进的 SEIR 模型示意图

同时考虑到非隔离的感染者 I 和潜伏者 E 对易感人群的影响，又有隔离解除的易感者 s_q 重新转变为 S ，因此易感者人数控制方程为：

$$\frac{dS}{dt} = -[c\beta + cq(1-\beta)]SI + kS_q$$

对于潜伏者人数的控制方程为：

$$\frac{dE}{dt} = c\beta(1-q)SI - \eta E$$

其中 η 为潜伏者向感染者的转化速率，取 $\eta=1/7$

对于感染者人数的控制方程为：

$$\frac{dI}{dt} = \eta E - (\delta_I + \alpha + \gamma_I)I$$

其中 δ_I 为感染者的隔离速率， α 为死亡率， γ_I 为感染者的恢复速率。

对于隔离易感者人数的控制方程为：

$$\frac{dS_q}{dt} = cq(1 - \beta)SI - \lambda S_q$$

对于隔离潜伏者人数的控制方程为：

$$\frac{dE_q}{dt} = c\beta qSI - \delta_q E_q$$

其中， δ_q 是隔离潜伏者转化成感染者的速率。

对于住院患者人数的控制方程为：

$$\frac{dH}{dt} = \delta_I I + I_q E_q - (\alpha + \gamma_H)H$$

其中， γ_H 为感染者的恢复速率。

对于痊愈者的控制方程为：

$$\frac{dR}{dt} = \gamma_I I + \gamma_H H$$

综上所述，得到以下方程组：

$$\left\{ \begin{array}{l} \frac{dS}{dt} = -[c\beta + cq(1 - \beta)]SI + kS_q \\ \frac{dE}{dt} = c\beta(1 - q)SI - \eta E \\ \frac{dI}{dt} = \eta E - (\delta_I + \alpha + \gamma_I)I \\ \frac{dS_q}{dt} = cq(1 - \beta)SI - \lambda S_q \\ \frac{dE_q}{dt} = c\beta qSI - \delta_q E_q \\ \frac{dH}{dt} = \delta_I I + \delta_q E_q - (\alpha + \gamma_H)H \\ \frac{dR}{dt} = \gamma_I I + \gamma_H H \end{array} \right.$$

分析过程与 SIR 模型类似。

5.2 传染数的模型建立

5.2.1 基本传染数的影响因素

由 5.1.2 可知, $R_0 = \frac{\lambda}{\mu}$, 根据该式, 我们可以分析得到如下三个因素对天然传播能力——基本传染数 R_0 的影响。

1. 感染周期

所谓感染周期, 是指一个病人从被感染到痊愈, 或者从被感染到死亡的时间跨度。显然, 疾病的感染周期越长, 患者接触健康人的机会就越多, R_0 也就越高。这样说来, 那些快速致人死亡的传染病虽然凶狠, 却不是传染性最强的。最典型的例子就是鼠疫, 因为宿主死亡太快, 导致鼠疫杆菌来不及传播到其他宿主体内。另一个极端代表就是艾滋病。众所周知, 艾滋病的潜伏期长达十年甚至更久。

2. 传播途径

不同疾病的传播途径不同, 导致了疾病传染的效率也大相径庭。艾滋病只能通过血液, 母婴和性方式传播, 所以大部分情况下是不会传给健康人的。而新冠肺炎可以通过飞沫传播, 所以人传人的现象非常普遍。不同的传播途径在接触频率一定的情况下, 会大幅影响有效接触的次数。

3. 易感者和传染者接触频率

在不同国家, 不同地区, 人口密度有很大差别。显然, 在人口密集的城市, 人与人接触的频率更高, 疾病也更容易传播; 在人烟稀少的地区, 人与人接触较少, 疾病的传播会受到一定阻碍。如: 截止至 6 月 12 日, 中国累计确诊病例为 84671 例, 西藏自治区累计确诊病例 1 例, 新疆维吾尔自治区累计病例 76 例。

5.2.2 实际传染数的影响因素

对于基本传染数的影响因素, 除了传播途径由病毒本身决定, 有如下几种途径会影响剩下的两种影响因素对基本传染数的贡献, 从而降低实际感染数。

1. 易感者比例

在感染初期, 对于一个传染者, 可接触的人群中全部都是易感者。随着传染系统的运行, 易感者比例的下降, 对于一个传染者, 可接触的人群中易感者比例下降, 传染进程变慢。故可得实际传染数正比于易感者比例。当易感者比例等于 0 时, 传染者不再可能感染易感者, 此时实际传染数为 0。即 $R \propto s(t)$ 。

2. 隔离

这里所说的隔离, 是一个广义的概念。首先可以是对患者的隔离。把患病者隔离开来, 能够减少甚至杜绝患者和健康人的接触。其次是对人群聚集的限制。一些表面健康的潜伏者也可能是疾病的感染者, 限制人群聚集可以减少潜在感染者和健康人的接触机会。同时提高卫生环境(如戴口罩或者洗手)也可以视作易感者与病毒的隔离。

根据三项统计的结论可知，增加洗手频率在高收入国家能够减少约 25% 的流感与感冒，伦敦的全城封锁减少了约 70% 的近距离接触。有 95% 的把握认为患者佩戴口罩降低新冠疫情的传染有 70% 的影响。所以假设洗手频率可让实际传染数最多降低 25%，封城降低社交距离能让实际传染数最多降低 70%，而佩戴口罩最多能使实际传染数最多降低 50%。即 $R \propto (1-x_1)(1-x_2)(1-x_3)$ 。

隔离出现症状的病例能将实际传染数降低最多 40%，而隔离潜伏者和没有症状的接触者能将实际传染数降低最多 50%。（注：隔离指定人群的效果和封城所得到的效果是存在很大差异的，不管是疫情防控效果，还是经济、心理疫后恢复效果）即 $R \propto (1-x_4)(1-x_5)$ 。

3. 温度

对新冠疫情来说，每增加一摄氏度会使实际传染数降低 1.2%。武汉市年温差约为 32 度，所以夏天相对于冬天的 R 值会降低 38%。纽约是年温差约为 26 度，夏天相对于冬天的 R 值会降低 31%。

4. 疫苗

疫苗是对健康人的保护。接种疫苗可以激活人体自身的免疫系统，从而产生有效的抗体。在这里我们假设疫苗对防止易感者传染是一定有效的。通过注射疫苗，可将易感者转移成痊愈者。通过降低易感者比例，从而降低实际感染数。此时实际传染数正比于易感者注射疫苗的比例，即 $R \propto (1-x_6)$ 。

5. 特效药

特效药是对患者的直接治愈，使得疾病的感染周期提前结束，既保护了感染者自身，也使疾病无法继续传播给其他人。需要注意的是，对于一种全新的流行病，疫苗和特效药都不太可能在短时间内研发出来。此外，人类发明的特效药基本是针对细菌的（抗生素），对于病毒几乎无能为力。故我们忽略特效药对新冠病毒的影响。

综上，分析以上因素对实际传染数的影响，可得实际传染数 R 为

$$R = \prod_{i=1}^7 (1-X_i) s(t) R_0$$

5.2.3 实际传染数的实际意义分析

实际传染数是指平均情况下在人为干预的情况下，一个病人在感染周期内能够传染的平均人数。故考虑离散情况下，在一个传染周期内，易感者转化为传染者（SEIR 模型为感染者）的数量为传染者人数的若干倍数，即 $i(t-1) - i(t) = R_0 * i(t)$ 此时相当于等比数列增长。将其连续化后，易知其为指数增长。 $R > 1$ 时，传染者人数会不断增加， $R = 1$ 时，传染者人数短时间内基本恒定， $R < 1$ 时，传染者人数会逐渐降低，直至该病消失。故当某种疾病的实际传染数 $R > 1$ 时，医学上将其划分为传染病。当 $R = 1$ 时，医学上将其划分为流

行病。当 $R < 1$ 时，医学上称人群对该疾病具有群体免疫。故 R 从大于 1 变为小于 1 的时间点为疫情拐点。

一个政府采取的所有措施的根本目的都是为了降低实际传染数 R 。将上式记为 $R = (1 - \alpha)R_0$ ，所有措施都只需保证降低 R_0 的一定比例 α 使 $R < 1$ 即可，此时 $\alpha > \left(1 - \frac{1}{R_0}\right)$ 。这也就意味着政府不需要断绝所有或几乎所有传染途径来达到阻止病情传染的效果。对新冠病毒而言 $R_0 \approx 2.5$ 时只需 $\alpha > 60\%$ 即可。故政府只需要减少 60% 的传染就可以达到群体免疫效果，进而到达疫情拐点。

5.2.4 考虑了隔离易感者、隔离潜伏者的 SEIR 模型的分析

我们采用了改进的 SEIR 模型，考虑了隔离易感者、隔离潜伏者和住院病人。通过模型分析，在使用感染率、接触率、个利率等变量的条件下，我们对潜伏者转化为感染者、隔离潜伏者转化为易感者、感染者康复时间等变量进行了假设（见 5.1.4），构造了相应的微分方程组，并且在 matlab 软件上编写了相应程序。

在程序当中，我们使用了湖北作为分析对象，对该省人口、1 月 23 至 1 月 29 日新增确诊病例数总数、感染者、正在被隔离的潜伏者（估计值）、正在住院的患者，为感染者和被隔离的潜伏者之和、官方公布的出院人数进行参数设置，在此基础上，从此数据后的第一天开始迭代，假设潜伏者不具有传染性，模拟预测各类型人数的变化情况的到了如下曲线图，

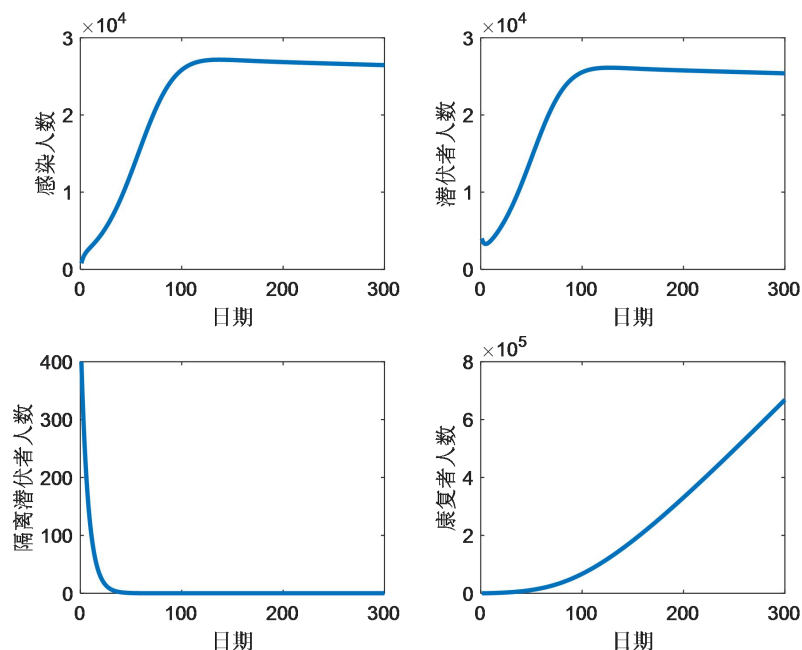


图 4 改进 SEIR 模型预测曲线

分析曲线可以发现，感染者在一个 S 型曲线的增长后达到平稳，即病毒实际传染数从大于 1 转变为小于 1，使得疫情得到了控制。

6 模型检验与比较

6.1 中国疫情分析

6.1.1 中国疫情拐点出现时间判断

根据疫情实时大数据报告⁷，我们选取了中国自 1 月 26 日到 2 月 29 日 35 天疫情高峰段的累计确诊数据。使用 matlab 结合 Logistics 曲线对数据进行拟合，横坐标选取为自 1 月 26 日起的间隔天数，纵坐标选取为当日累计确诊数。拟合结果如下：

其函数解析式为

$$y = \frac{81101}{1 + e^{3.1865 - 0.2230x}}$$

通过对该函数求二阶导数的零点，解得函数拐点为 14.29。对应日期为 2 月 8 日。

6.1.2 中国疫情拐点成因判断

取 $R_0=5$ 。政府提高武汉的卫生条件，全国对武汉输送卫生应急物资，取 $x_1=20\%$ 。政府对疫情爆发地武汉进行封城隔离，取 $x_2=70\%$ 。政府呼吁减少外出并佩戴口罩，取 $x_3=20\%$ （封城隔离后社交减少，口罩的作用相对降低）。武汉 2 月 8 日病例数为 37251，武汉总人口为 1400 万人，易感者比例 $s(t)/N=99.7\%$ 。此外，忽略其他因素的影响，即 $x_4 = x_5 = x_6 = x_7 = 0$ ；故得出对应实际传染数 $R=0.96 < 1$ 。

6.2 美国疫情分析

6.2.1 美国疫情拐点成因判断

疫情的拐点出现为 $R=1$ 时，根据实际传染数 R 的公式可知。在一定的基本传染数 R_0 下，存在两种原因使得疫情出现拐点。

- 1) 采取严格的防疫措施, 降低 $\prod_{i=1}^7 (1-x_i)$ 的值, 使得 $R=1$ 。
- 2) 持续当前的防疫措施, 随着疫情的传播, 降低 $s(i)$ 的值, 使得 $R=1$ 。

6.2.2 美国以轻拐点出现时间预测

对于第一种情况, 当政府采取严格防疫措施时, 如果 $\prod_{i=1}^7 (1-x_i) \leq \frac{1}{s(i)R_0}$ 时,

疫情出现拐点。故这种情况取决于政府采取防疫措施的严格程度与民众遵从防疫措施的程度。

对于第二种情况, 我们根据四种模型和美国 3 月 12 日至 6 月 20 日一百天每日累计确诊数据, 使用蒙特卡洛最小二乘法, 用差商近似代替导数, 随机撒点 10000 次, 计算出最接近的模型参数, 并据此递推接下来的疫情走向并预测疫情拐点。

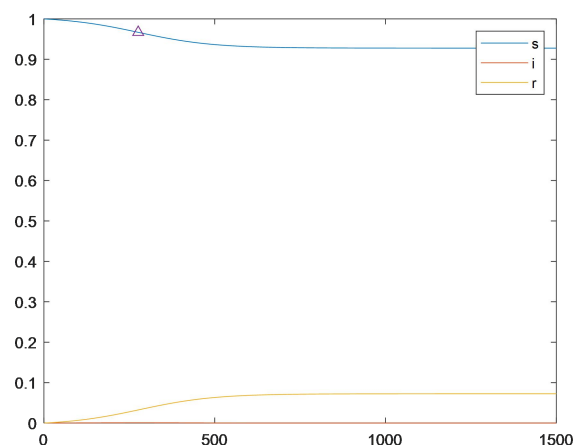


图 5 SIR 模型预测结果

1) SIR 模型拟合标准差为 0.00034, 预测疫情拐点为第 276 天, 对应日期为 2020 年 12 月 13 日, 此时累计 3.32% 的人被感染。疫情结束后有累计 7.23% 的人被感染并具有抗体。

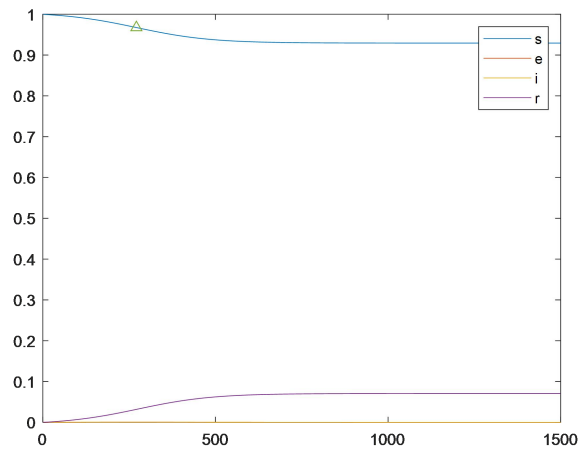


图 6 SEIR 模型预测结果

2) SEIR 模型拟合标准差为 0.00032，预测疫情拐点为第 271 天，对应日期为 2020 年 12 月 8 日，此时累计 3.62% 的人被感染。疫情结束后有累计 7.06% 的人被感染并具有抗体。

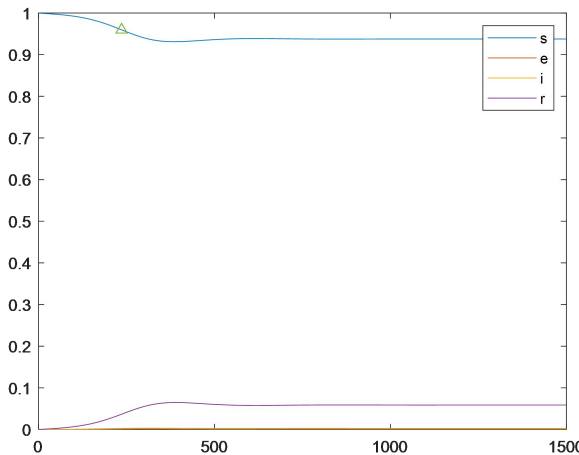


图 7 SEIRS 模型预测结果

3) SEIRS 模型拟合标准差为 0.00042，预测疫情拐点为第 236 天，对应日期为 2020 年 11 月 3 日，此时累计 4.04% 的人被感染。疫情结束后有 5.87% 的人具有临时抗体。

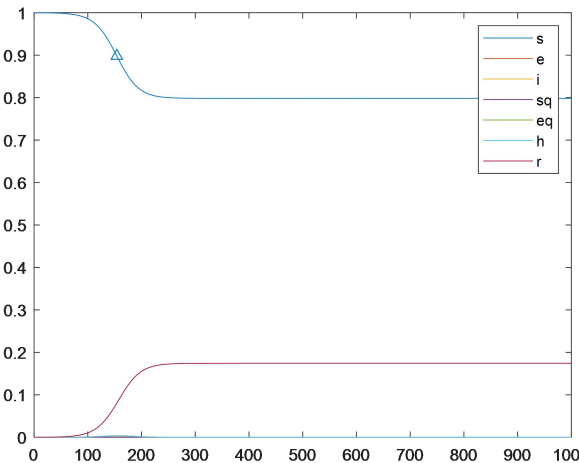


图 8 改进 SEIR 模型结果

4) 改进 SEIR 模型拟合标准差为 0.0017, 预测疫情拐点为 154 天, 对应日期为 2020 年 8 月 13 日。此时累计 10.11% 的人被感染, 疫情结束后累计 17.43% 的人被感染并具有抗体。

综上结果, 前三个模型拟合的标准差比改进 SEIR 模型小一个量级。说明拟合效果更好。前三个模型给出的拐点预测时间基本属于 11 月中旬到 12 月中旬的时间段, 预测总感染人数基本为 5% 到 7% (SEIRS 的结果为具有临时抗体的人数比例)。改进 SEIR 模型给出的拐点预测时间为 8 月份中旬, 预测总感染人数为 17%。

7 模型评价

7.1 模型的优点

SEIRS 的模型是由具体病情抽象出来的, 由于存在潜伏期, 故引入了感染者人群。由于存在部分痊愈者再次感染病毒, 故而引入了痊愈者向易感者转移的过程。较为凝练准确地刻画了病毒传染系统。

7.2 模型的缺点及改进方向

1. R 的模型考虑并不完全, 可以考虑更多的影响指标来更好的模拟实际情况, 例如: 洗手频率, 平均肢体距离, 检测频率等等。

2. 未考虑超级传染者的存在性, 如一传十, 一传百等极端情况, 增设该人群会加速易感者被感染的速度。

3. 未考虑动物对人的传染, 新冠病毒可能是由野生蝙蝠带来的。此外, 也出现了部分动物感染的病例, 这些动物也可能按照一定的基本传染数传染人类, 从而加快易感者被感染的速度。

4. 传染者痊愈过程刻画不够真实, 传染者人群在接受治疗时, 由于病人的支付能力、病床容量、救助物资量和医护人员数量会导致传染者转移至痊愈者的过程与本文模型不符。

5. 未考虑区域与人口流动对疫情传播的影响。不同地区的人口密度、气候环境不同, 一定程度上影响了实际传染数的大小。同时不同地区存在人口迁移, 需要对不同地区内部单独计算 SEIRS 模型, 然后使用马尔科夫链模型计算不同地区人口迁移。

8 参考文献

- [1] Neil M Ferguson, Daniel Laydon, Gemma Nedjati-Gilani, Natsuko Imai, Kylie Ainslie, Marc Baguelin, Sangeeta Bhatia, Adhiratha Boonyasiri, Zulma Cucunubá, Gina Cuomo-Dannenburg, Amy Dighe, Ilaria Dorigatti, Han Fu, Katy Gaythorpe, Will Green, Arran Hamlet, Wes Hinsley, Lucy C Okell, Sabine van Elsland, Hayley Thompson, Robert Verity, Erik Volz, Haowei Wang, Yuanrong Wang, Patrick GT Walker, Caroline Walters, Peter Winskill, Charles Whittaker, Christl A Donnelly, Steven Riley, Azra C Ghani. Report 9 - Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand
<http://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/covid-19/report-9-impact-of-npis-on-covid-19/>
- [2] Swicochev, SEIR 和 SEIRS 模型
<https://www.bilibili.com/read/cv5559283>
- [3] 程序员小灰, 漫画: 什么是流行病的 R0 和 R ?
<https://blog.csdn.net/csdnnews/article/details/105631437>
- [4] 酱紫君, 传染病的基本数学模型——知乎
<https://www.zhihu.com/tardis/sogou/qus/367466399>
- [5] Denny·Z, 基于改进的 SEIR 模型预测新冠肺炎(COVID-19)疫情
<https://zhuanlan.zhihu.com/p/113768548>
- [6] 疫情实时大数据报告
https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_pc_3#tab4
- [7] 疫情实时大数据报告
https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_pc_3#tab4

9 附件

9.1 曲线拟合判断疫情拐点程序 (MATLAB 编写)

```
y=[2761 4535 5997 7736 9720 11821 14411 17238 20471 24363 28060 31211 34598
37251 40235 42708 44730 59882 63932 66576 68584 70635 72528 74279 75101
75993 76392 77041 77262 77779 78190 78630 78959 79389 79968];

[n,m]=size(y);

x=1:m;

myfittype=fittype('a/(1+exp(b+c*x))','dependent',{'y'},'independent',{'x'},'coefficients',
{'a','b','c'});

myfit=fit(x,y,myfittype,'StartPoint',[0,0,0])
```

```
plot(myfit,x,y);  
  
hold on  
  
syms x;  
  
a=myfit.a;  
  
b=myfit.b;  
  
c=myfit.c;  
  
y=a./(1+exp(b+c.*x));  
  
y2=diff(y,x,2);  
  
x0=solve(y2,x);  
  
plot(x0,subs(y,x,x0),'^');
```

9.2 SIR 模型拟合预测程序（MATLAB 编写）

```
y=[1264 1322 1678 3499 4629 5894 9345 14250 19624 22043 32717 46332 53268  
65285 83507 101657 121117 139675 161367 184183 206207 236339 270473 312076  
334125 363349 394182 425828 463433 498674 530384 559245 586941 610632  
641397 674829 710021 738697 762496 789383 825306 844992 877497 916348  
955488 985060 1004942 1029878 1056646 1092656 1125305 1156744 1185167  
1209702 1234592 1256639 1289028 1318686 1342723 1365308 1381665 1406519  
1427587 1453381 1480975 1503684 1526134 1532861 1555133 1573778 1597130  
1627409 1650677 1672527 1689727 1711569 1730595 1750377 1772643 1797949  
1822117 1842243 1862879 1887708 1907840 1928026 1973883 1995854 2012001  
2030323 2052816 2072274 2094368 2123102 2150245 2166685 2188198 2215580  
2242914 2275218];  
[n,day]=size(y);  
N=3.3e8;  
y=y/N;  
% lambda=0.2586;  
% w=0.07;  
% miu=0.0821;  
  
sum=0;  
minsum=100000;  
alambda=0;  
amiu=0;  
for tot=1:50000  
    tot;
```

```
s=[0.9998];
i=[0.0002];
r=[0];
ir=[0.0002];
lambda=rand();
miu=rand();
sum=0;
for o=2:day
    s=[s s(o-1)-lambda*i(o-1)*s(o-1)];
    i=[i i(o-1)+lambda*i(o-1)*s(o-1)-miu*i(o-1)];
    r=[r r(o-1)+miu*i(o-1)];
    ir=[ir r(o)+i(o)];
    sum=sum+(y(o)-ir(o))^2;
end
if(minsum>sum)
    minsum=sum;
    alambda=lambda;
    amiu=miu;
end
end
s=[0.9998];
e=[0];
i=[0.0002];
r=[0];
ir=[0.0002];
Day=day*15;
for o=2:Day
    s=[s s(o-1)-alambda*i(o-1)*s(o-1)];
    i=[i i(o-1)+alambda*i(o-1)*s(o-1)-amiu*i(o-1)];
    r=[r r(o-1)+amiu*i(o-1)];
end
sqrt(minsum/day)
plot(1:Day,s);
hold on
plot(1:Day,i);
hold on
plot(1:Day,r);
hold on
s2=diff(s,2);
for o=2:Day
    if(s2(o-1)<0&& s2(o)>0)
        plot(o,s(o),'^');
        break
    end
end
```

```
end
legend('s','i','r')
```

9.3 SEIR 模型拟合预测程序（MATLAB 编写）

```
y=[1264 1322 1678 3499 4629 5894 9345 14250 19624 22043 32717 46332 53268
65285 83507 101657 121117 139675 161367 184183 206207 236339 270473 312076
334125 363349 394182 425828 463433 498674 530384 559245 586941 610632
641397 674829 710021 738697 762496 789383 825306 844992 877497 916348
955488 985060 1004942 1029878 1056646 1092656 1125305 1156744 1185167
1209702 1234592 1256639 1289028 1318686 1342723 1365308 1381665 1406519
1427587 1453381 1480975 1503684 1526134 1532861 1555133 1573778 1597130
1627409 1650677 1672527 1689727 1711569 1730595 1750377 1772643 1797949
1822117 1842243 1862879 1887708 1907840 1928026 1973883 1995854 2012001
2030323 2052816 2072274 2094368 2123102 2150245 2166685 2188198 2215580
2242914 2275218];
[n,day]=size(y);
N=3.3e8;
y=y/N;
% lambda=0.2586;
% w=0.07;
% miu=0.0821;

sum=0;
minsum=100000;
alambda=0;
aw=0;
amiu=0;
for tot=1:10000
    tot;
    s=[0.9998];
    e=[0];
    i=[0.0002];
    r=[0];
    ir=[0.0002];
    lambda=rand()*2;
    w=rand();
    miu=rand();
    sum=0;
    for o=2:day
        s=[s s(o-1)-lambda*i(o-1)*s(o-1)];
        e=[e e(o-1)+lambda*i(o-1)*s(o-1)-w*e(o-1)];
        i=[i i(o-1)+w*e(o-1)-miu*i(o-1)];
```

```
    r=[r r(o-1)+miu*i(o-1)];
    ir=[ir r(o)+i(o)];
    sum=sum+(y(o)-ir(o))^2;
end
if(minsum>sum)
    minsum=sum;
    alambda=lambda;
    aw=w;
    amiu=miu;
end
end
s=[0.9998];
e=[0];
i=[0.0002];
r=[0];
ir=[0.0002];
Day=day*15;
for o=2:Day
    s=[s s(o-1)-alambda*i(o-1)*s(o-1)];
    e=[e e(o-1)+alambda*i(o-1)*s(o-1)-aw*e(o-1)];
    i=[i i(o-1)+aw*e(o-1)-amiu*i(o-1)];
    r=[r r(o-1)+amiu*i(o-1)];
end
sqrt(minsum/day)
plot(1:Day,s);
hold on
plot(1:Day,e);
hold on
plot(1:Day,i);
hold on
plot(1:Day,r);
hold on
s2=diff(s,2);
for o=day:Day
    if(s2(o-1)<0&& s2(o)>0)
        plot(o,s(o),'^');
        break
    end
end
end
legend('s','e','i','r')
```

9.4 SEIRS 模型拟合预测程序（MATLAB 编写）

```

y=[1264 1322 1678 3499 4629 5894 9345 14250 19624 22043 32717 46332 53268
65285 83507 101657 121117 139675 161367 184183 206207 236339 270473 312076
334125 363349 394182 425828 463433 498674 530384 559245 586941 610632
641397 674829 710021 738697 762496 789383 825306 844992 877497 916348
955488 985060 1004942 1029878 1056646 1092656 1125305 1156744 1185167
1209702 1234592 1256639 1289028 1318686 1342723 1365308 1381665 1406519
1427587 1453381 1480975 1503684 1526134 1532861 1555133 1573778 1597130
1627409 1650677 1672527 1689727 1711569 1730595 1750377 1772643 1797949
1822117 1842243 1862879 1887708 1907840 1928026 1973883 1995854 2012001
2030323 2052816 2072274 2094368 2123102 2150245 2166685 2188198 2215580
2242914 2275218];
[n,day]=size(y);
N=3.3e8;
y=y/N;
% lambda=0.2586;
% w=0.07;
% miu=0.0821;

sum=0;
minsum=100000;
alambda=0;
aw1=0;
aw2=0;
amiu=0;
for tot=1:10000
    tot;
    s=[0.9998];
    e=[0];
    i=[0.0002];
    r=[0];
    ir=[0.0002];
    lambda=rand();
    w1=rand();
    miu=rand();
    w2=rand();
    sum=0;
    for o=2:day
        s=[s s(o-1)-lambda*i(o-1)*s(o-1)+w2*r(o-1)];
        e=[e e(o-1)+lambda*i(o-1)*s(o-1)-w1*e(o-1)];
        i=[i i(o-1)+w1*e(o-1)-miu*i(o-1)];
        r=[r r(o-1)+miu*i(o-1)-w2*r(o-1)];
    end
    sum=sum+abs(s-0.9998)+abs(e-0)+abs(i-0.0002)+abs(r-0)+abs(ir-0.0002);
    if sum<minsum
        alambda=lambda;
        aw1=w1;
        aw2=w2;
        amiu=miu;
        minsum=sum;
    end
end

```



```
    ir=[ir r(o)+i(o)];
    sum=sum+(y(o)-ir(o))^2;
end
if(minsum>sum)
    minsum=sum;
    alambda=lambda;
    aw1=w1;
    amiu=miu;
    aw2=w2;
end
end
sqrt(minsum/day)
s=[0.9998];
e=[0];
i=[0.0002];
r=[0];
ir=[0.0002];
Day=day*15;
for o=2:Day
    s=[s s(o-1)-alambda*i(o-1)*s(o-1)+aw2*r(o-1)];
    e=[e e(o-1)+alambda*i(o-1)*s(o-1)-aw1*e(o-1)];
    i=[i i(o-1)+aw1*e(o-1)-amiu*i(o-1)];
    r=[r r(o-1)+amiu*i(o-1)-aw2*r(o-1)];
end
plot(1:Day,s);
hold on
plot(1:Day,e);
hold on
plot(1:Day,i);
hold on
plot(1:Day,r);
hold on
s2=diff(s,2);
for o=day:Day-2
    if(s2(o-1)<0&& s2(o)>0)
        plot(o,s(o),'^');
        break
    end
end
end
legend('s','e','i','r')
```

9.5 改进 SEIR 模型拟合预测程序（MATLAB 编写）

```

y=[1264 1322 1678 3499 4629 5894 9345 14250 19624 22043 32717 46332 53268
65285 83507 101657 121117 139675 161367 184183 206207 236339 270473 312076
334125 363349 394182 425828 463433 498674 530384 559245 586941 610632
641397 674829 710021 738697 762496 789383 825306 844992 877497 916348
955488 985060 1004942 1029878 1056646 1092656 1125305 1156744 1185167
1209702 1234592 1256639 1289028 1318686 1342723 1365308 1381665 1406519
1427587 1453381 1480975 1503684 1526134 1532861 1555133 1573778 1597130
1627409 1650677 1672527 1689727 1711569 1730595 1750377 1772643 1797949
1822117 1842243 1862879 1887708 1907840 1928026 1973883 1995854 2012001
2030323 2052816 2072274 2094368 2123102 2150245 2166685 2188198 2215580
2242914 2275218];
[n,day]=size(y);
N=3.3e8;
y=y/N;
minsum=10000;
ac=0;
adeltaI=0;
adeltaq=0;
agammaI=0;
agammaH=0;
abeta=0;
aq=0;
aalpha=0;
ak=0;
aeta=0;

for i=1:30000
S1=0.999996;
E1=0;% 美国
I1=0.000004;%感染者
Sq1=0;%尚在接受医学观察的人数
Eq1=0;%估计值，为正在被隔离的潜伏者
H1=0;%正在住院的患者，为感染者和被隔离的潜伏者之和
R1=0;%官方公布的出院人数

%模型参数设定
sum=0;
c=5*rand(1);%接触率
deltaI=rand(1);%感染者的隔离速度
deltaq=rand(1);%隔离潜伏者向隔离感染者的转化速率
gammaI=rand(1);%感染者的恢复率

```

```

gammaH=rand(1);%隔离感染者的恢复速率
beta=rand(1);%传染概率
q=rand(1);%隔离比例
alpha=rand(1);%病死率
k=rand(1);%隔离接触速度，为14天的倒数
eta=rand(1);%潜伏者向感染者的转化速度，平均潜伏期为7天，为7天的倒数
%差分迭代方程
for idx =2:day %不考虑潜伏者传染情况
    S1=[S1 S1(idx-1)-(c*beta+c*q*(1-beta))*S1(idx-1)*(I1(idx-1))+k*Sq1(idx-1)];%易感人数迭代
    E1=[E1 E1(idx-1)+c*beta*(1-q)*S1(idx-1)*I1(idx-1)-eta*E1(idx-1)];%潜伏者人数迭代
    I1=[I1 I1(idx-1)+eta*E1(idx-1)-(deltaI+alpha+gammaI)*I1(idx-1)];%感染者人数迭代
    Sq1=[Sq1 Sq1(idx-1)+c*q*(1-beta)*S1(idx-1)*I1(idx-1)-k*Sq1(idx-1)];%隔离易感感染着人数迭代
    Eq1=[Eq1 Eq1(idx-1)+c*beta*q*S1(idx-1)*I1(idx-1)-deltaq*Eq1(idx-1)];%隔离潜伏者人数迭代
    H1=[H1 H1(idx-1)+deltaI*I1(idx-1)+deltaq*Eq1(idx-1)-(alpha+gammaH)*H1(idx-1)];%住院患者人数迭代
    R1=[R1 R1(idx-1)+gammaI*I1(idx-1)+gammaH*H1(idx-1)];%康复人数迭代
    sum=sum+(I1(idx)+H1(idx)+R1(idx)-y(idx))^2;
end
if (minsum>sum)
    minsum=sum;
    ac=c;%接触率
    adeltaI=deltaI;%感染者的隔离速度
    adeltaq=deltaq;%隔离潜伏者向隔离感染者的转化速率
    agammaI=gammaI;%感染者的恢复率
    agammaH=gammaH;%隔离感染者的恢复速率
    abeta=beta;%传染概率
    aq=q;%隔离比例
    aalpha=alpha;%病死率
    ak=k;%隔离接触速度，为14天的倒数
    aeta=eta;
end
end
c=ac;
deltaI=adeltaI;
deltaq=adeltaq;
gammaI=agammaI;
gammaH=agammaH;
beta=abeta;
q=aq;

```

```

alpha=aalpha;
k=ak;
eta=aeta;
Day=day*10;
S1=0.999996;
E1=0;% 美国
I1=0.000004;%感染者
Sq1=0;%尚在接受医学观察的人数
Eq1=0;%估计值，为正在被隔离的潜伏者
H1=0;%正在住院的患者，为感染者和被隔离的潜伏者之和
R1=0;%官方公布的出院人数
Sum=0;
for idx =2:Day %不考虑潜伏者传染情况
    S1=[S1 S1(idx-1)-(c*beta+c*q*(1-beta))*S1(idx-1)*(I1(idx-1))+k*Sq1(idx-1)];%
    易感人数迭
    E1=[E1 E1(idx-1)+c*beta*(1-q)*S1(idx-1)*I1(idx-1)-eta*E1(idx-1)];%潜伏者人
    数迭代
    I1=[I1 I1(idx-1)+eta*E1(idx-1)-(deltaI+alpha+gammaI)*I1(idx-1)];%感染者人数
    迭代
    Sq1=[Sq1 Sq1(idx-1)+c*q*(1-beta)*S1(idx-1)*I1(idx-1)-k*Sq1(idx-1)];%隔离易
    感染着人数迭代
    Eq1=[Eq1 Eq1(idx-1)+c*beta*q*S1(idx-1)*I1(idx-1)-deltaq*Eq1(idx-1)];%隔离潜
    伏者人数迭代
    H1=[H1 H1(idx-1)+deltaI*I1(idx-1)+deltaq*Eq1(idx-1)-(alpha+gammaH)*H1(idx-
    1)];%住院患者人数迭代
    R1=[R1 R1(idx-1)+gammaI*I1(idx-1)+gammaH*H1(idx-1)];%康复人数迭代
end
sqrt(minsum/day)
plot(1:Day,S1);
hold on
plot(1:Day,E1);
hold on
plot(1:Day,I1);
hold on
plot(1:Day,Sq1);
hold on
plot(1:Day,Eq1);
hold on
plot(1:Day,H1);
hold on
plot(1:Day,R1);
hold on

s2=diff(S1,2);

```

```
for o=day:Day
    if(s2(o-1)<0&&S2(o)>0)
        plot(o,S1(o),'^');
        break
    end
end
legend('s','e','i','sq','eq','h','r');
```