

下一代矩阵法计算基本再生数的 MATLAB 实现

1 功能预览

1.1 基本操作

输入模型对应的常微分方程，运行程序，输出该模型对应的基本再生数的公式。进一步输入各个参数值，代码将进一步给出带入后的基本再生数数值。

对于不同的模型，使用者需要修改的只有：1) 模型的微分方程部分，2) 需要带入的参数值部分。

1.2 结果展示

$$\text{eigMat} = (S \cdot \beta \cdot \kappa \cdot \omega \cdot p) / ((dr + \gamma) \cdot (dr - \omega \cdot p + \omega + \omega \cdot p)) - (S \cdot \beta \cdot \omega \cdot (p - 1)) / ((dr + f + \gamma) \cdot (dr - \omega \cdot p + \omega + \omega \cdot p))$$

$R_0 = 2.910937500000000$

2 下载并安装 MATLAB

2.1 注册并下载

点击下面的超链接访问 MathWorks 官网，用厦大邮箱注册 MathWorks 账号，登录并下载 MATLAB (学校已购买 MATLAB 的学术使用权)

[下载 MATLAB、Simulink、Stateflow 和其他 MathWorks 产品](#)

若超链接无效，则可复制如下链接至浏览器：

https://ww2.mathworks.cn/downloads/web_downloads?s_iid=hp_ff_t_downloads

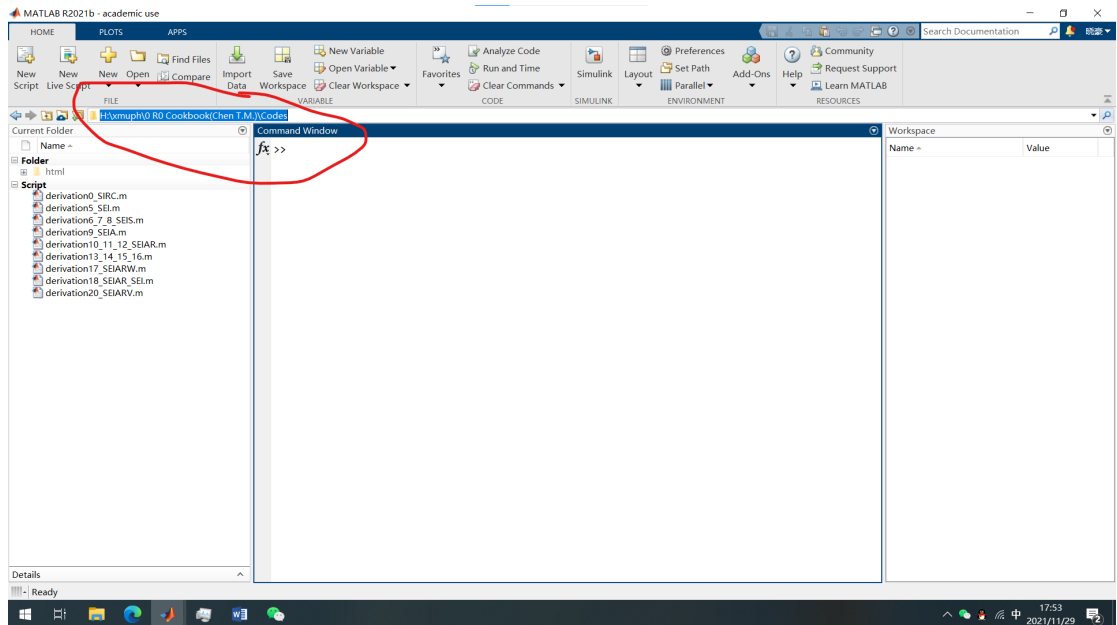
2.2 安装 MATLAB

使用厦大邮箱注册的账号登录，选择学校的许可证，勾选安装所有的工具箱。安装完成后根据提示进行激活。

3 在 MATLAB 中运行代码

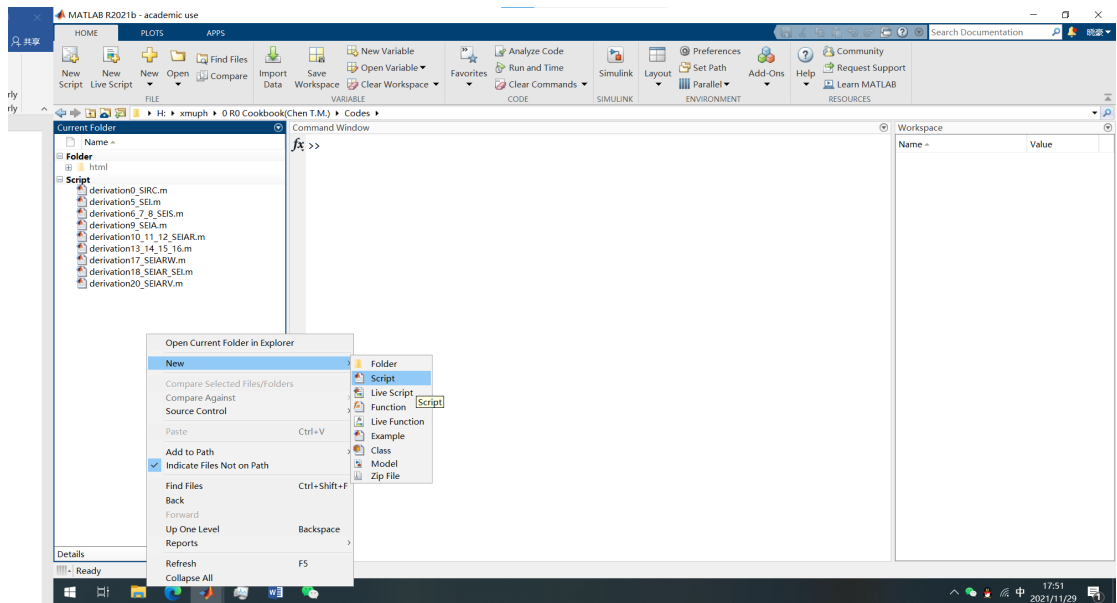
3.1 选择当前文件夹

运行 MATLAB，在图示位置输入路径，选择当前文件夹



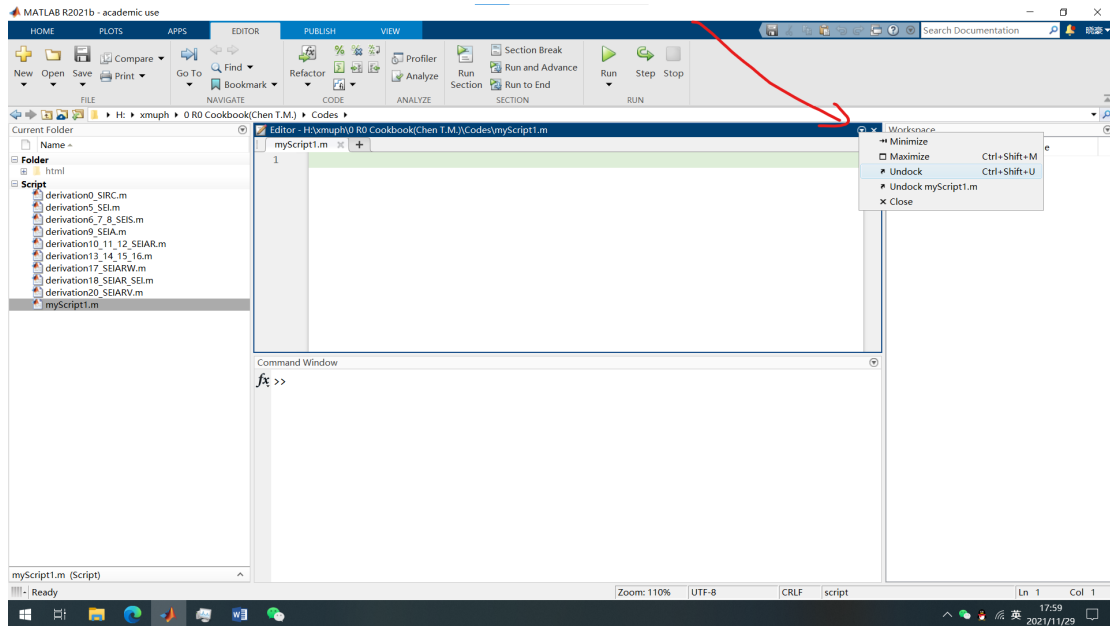
3.2 新建脚本

在当前文件夹（最左边的一栏）的空白处右键，选择新建脚本，并将它重命名为 myScript1.m



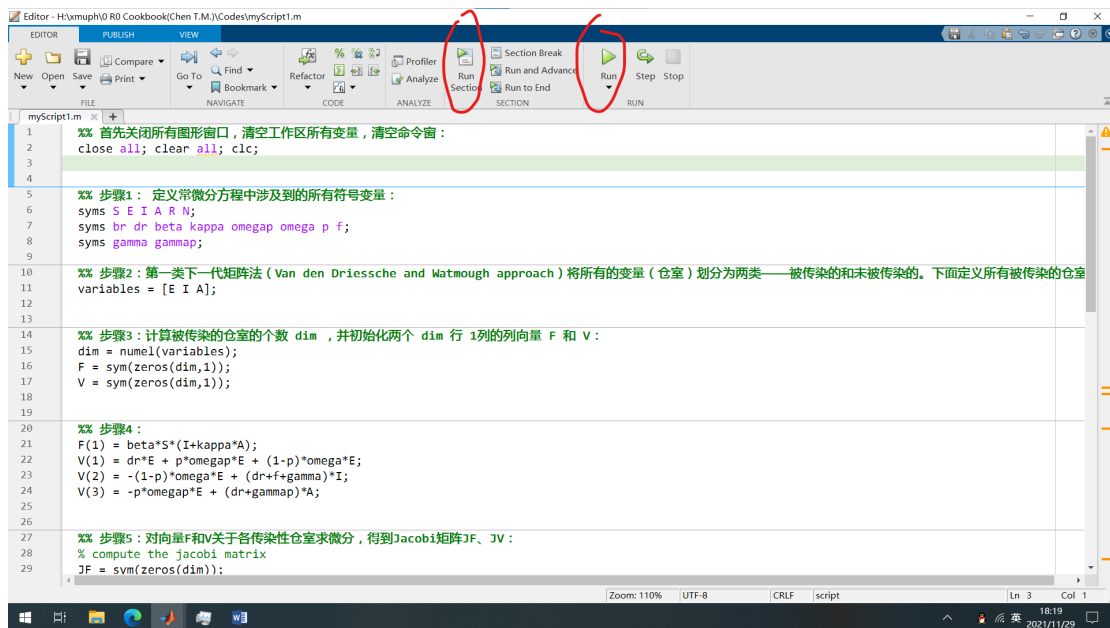
3.3 取消停靠

点击箭头所示的按钮，选择取消停靠



3.4 运行代码

在脚本中输入代码，使用分节符%%，即可运行或逐节运行。



4 下一代矩阵法的 MATLAB 实现

对于不同的问题，我们只需要修改下列步骤中的步骤 1，2，4，10 中和模型有关的部分。
例如，为了计算下图所示的 SEIAR 模型的基本再生数

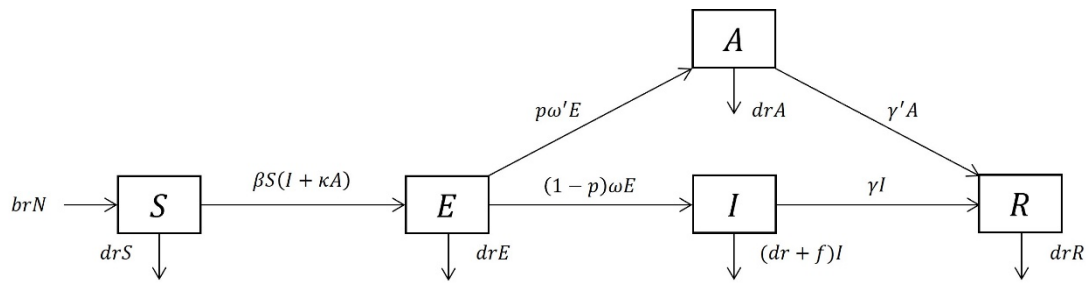


图 1

我们将下面的文本复制到刚刚新建的脚本中(从下一行开始，直到本文档结束的所有文本)，并点击运行。

%% 首先关闭所有图形窗口，清空工作区所有变量，清空命令窗：

```
close all; clear all; clc;
```

%% 步骤 1： 定义常微分方程中涉及到的所有符号变量：

```
syms S E I A R N;
```

```
syms br dr beta kappa omegap omega p f;
```

```
syms gamma gammap;
```

%% 步骤 2： 第一类下一代矩阵法（Van den Driessche and Watmough approach）将所有的变量（仓室）划分为两类——被传染的和未被传染的。下面定义所有被传染的仓室：

```
variables = [E I A];
```

%% 步骤 3: 计算被传染的仓室的个数 dim，并初始化两个 dim 行 1 列的列向量 F 和 V：

```
dim = numel(variables);
```

```
F = sym(zeros(dim,1));
```

```
V = sym(zeros(dim,1));
```

%% 步骤 4:

```
F(1) = beta*S*(I+kappa*A);
```

```
V(1) = dr*E + p*omegap*E + (1-p)*omega*E;
```

```
V(2) = -(1-p)*omega*E + (dr+f+gamma)*I;
```

```
V(3) = -p*omegap*E + (dr+gammap)*A;
```

%% 步骤 5: 对向量 F 和 V 关于各传染性仓室求微分, 得到 Jacobi 矩阵 JF、JV:

```
% compute the jacobi matrix
```

```
JF = sym(zeros(dim));
```

```
JV = sym(zeros(dim));
```

```
for i = 1:dim
```

```
    for j = 1:dim
```

```
        JF(i,j) = diff(F(i),variables(j));
```

```
        JV(i,j) = diff(V(i),variables(j));
```

```
    end
```

```
end
```

%% 步骤 6: 计算并输出下一代矩阵 Mat:

```
invJV = inv(JV)
```

```
Mat = JF*invJV
```

%% 步骤 7: 有效再生数被定义为“下一代矩阵的最大特征值”。计算下一代矩阵的特征值:

```
eigMat = eig(Mat)
```

%% 步骤 8: 找出上一步给出的特征值中最大的一个, 即为有效再生数 Reff.

%% 步骤 9: 将无病平衡点(S, E, I, A, R) = (N, 0, 0, 0, 0)带入 Reff 的表达式中, 即得到 R0.

%% 步骤 10: 给参数赋值

```
s = 1e7;  
beta = 9e-8;  
omega = 1/5;  
omegap = 1/7;  
gamma = 1/3;  
gammap = 1/5;  
kappa = 0.8;  
p = 0.3;  
br = 0;  
dr = 0;  
f = 0;
```

%% 步骤 11: 将参数的值带入 Reff 的表达式

```
R0 = eval(eigMat(end))
```