

《系统仿真与 matlab》综合试题

题 目： 弱肉强食问题—Volterra 模型

编 号： 12

难度系数：

姓 名 杨毅远

班 级 自动化实验班 1501 班

学 号 U201514417

联系方式 15927088163

成 绩

目录

《系统仿真与 matlab》综合试题.....	1
系统建模与仿真研究报告.....	1
一. 问题模型介绍.....	1
弱肉强食问题—Volterra 模型.....	1
二. 试题建模过程.....	2
1. 题目理解:	2
2. 做题目前的准备工作:	2
3. 系统中只考虑羊群独立生长(食饵独立生长):	3
4. 系统中存在捕食者(狼群), 但是其数量不变化:	3
5. 系统中存在捕食者(狼群), 且其数量随食饵(羊群)变化:	3
6. 系统在 5 的前提下, 考虑种群的自身阻滞作用:	3
7. 系统在 6 的前提下, 考虑人工捕获:	4
8. 再考虑羊的食物——草, 对于狼和羊的数量的影响:	4
9. 考虑 Volterra 模型下的系统稳定性情况:	4
三. 系统仿真流程.....	9
四. 系统仿真关键点.....	9
1. MATLAB 仿真界面的设计	9
2. 微分方程组处理.....	9
3. 考虑实际情况, 对于数据进行合理的约定.....	10
4. 仿真参数的输入.....	11
5. 仿真对象的选择.....	11
6. 仿真模式的选择.....	11
7. 三种群仿真.....	11
8. 图形动态仿真.....	11
9. 普通二维仿真.....	12
10. 图像保存.....	12
五. 程序运行指南.....	12
1. 关于文件及内容.....	12
2. 操作系统介绍.....	13
六. 系统模型的亮点以及新颖的地方.....	13
七. 程序运行实例分析.....	14
八. 心得体会.....	21
九. 参考书籍及资料.....	21
十. 部分代码附录(手写, 非 GUI 生成).....	22

系统建模与仿真研究报告

——自实 1501 班 杨毅远 U201514417

一. 问题模型介绍

弱肉强食问题—Volterra 模型

处于同一自然环境中两个种群之间的关系除了相互竞争和相互依存之外，还有一种更为有趣的生存方式：种群甲靠丰富的天然资源生长，而种群乙则靠掠食甲为生。地中海里的食用鱼与鲨鱼，加拿大森林中的美洲兔与山猫，阿尔卑斯山中的落叶松与芽虫等都是这种生存方式的典型。生态学上种群甲称为食饵 (Prey)，种群乙称为捕食者(Predator)，二者共处组成食饵—捕食者系统(简称 P—P 系统)。

模型假设

食饵和捕食者在时刻 t 的数量分别记作 $x_1(t)$ 和 $x_2(t)$ ，因为大海中资源丰富，可以假设如果食饵独立生存则将以增长率 r_1 按指数规律增长，即有 $\dot{x}_1 = r_1 x_1$ 。捕食者的存在使食饵的增长率降低，设降低的程度与捕食者数量成正比。于是 $x_1(t)$ 满足方程 $\dot{x}_1(t) = x_1(r_1 - \lambda_1 x_2)$ 比例系数 λ_1 ，反映捕食者掠取食饵的能力。

捕食者离开食饵无法生存，若设它独自存在时死亡率为 r_2 ，即 $\dot{x}_2 = -r_2 x_2$ ，而食饵为它提供食物的作用相当于使死亡率降低、或使之增长。设这个作用与食饵数量成正比，于是 $x_2(t)$ 满足 $\dot{x}_2(t) = x_2(-r_2 + \lambda_2 x_1)$ 比例系数 λ_2 ，反映食饵对捕食者的供养能力。

上述方程是在没有人工捕获情况下自然环境中食饵与捕食者之间的制约关系，是 Volterra 提出的最简单的模型。可以看出这个模型没有考虑自身的阻滞作用。

仿真要求 系统输入为仿真时间 T ，食饵和捕食者在初始时刻的数量 $x_1(0)$ 和 $x_2(0)$ ，食饵独立生存增长率 r_1 ，捕食者掠取食饵的能力 λ_1 ，捕食者独自存在时死亡率 r_2 ，食饵对捕食者的供养能力 λ_2 ，系统输出为食饵和捕食者在时刻 t 的数量 $x_1(t)$ 和 $x_2(t)$ 。要求有输入、输出界面及仿真过程。

二. 试题建模过程

1. 题目理解:

根据高中生物所学以及平时日常知识的积累,很容易理解题目的要求。即建立一个存在捕食者与被捕食者(食饵)的系统,以时间为线,查看不同参数下,捕食者和食饵的数量关系的变化情况。步骤应该如下,首先根据自己建立的模型求解出一组能使系统稳定的系数,即捕食者与被捕食者的数量是呈现周期变化的。再用 MATLAB 工具的一系列画图函数以及动画效果,实际模拟其数量的变化情况,在考虑实际情况的因素下,看仿真结果是否与实际情况相符,若相符则模型建立基本正确,若不成功就需要再改进模型。

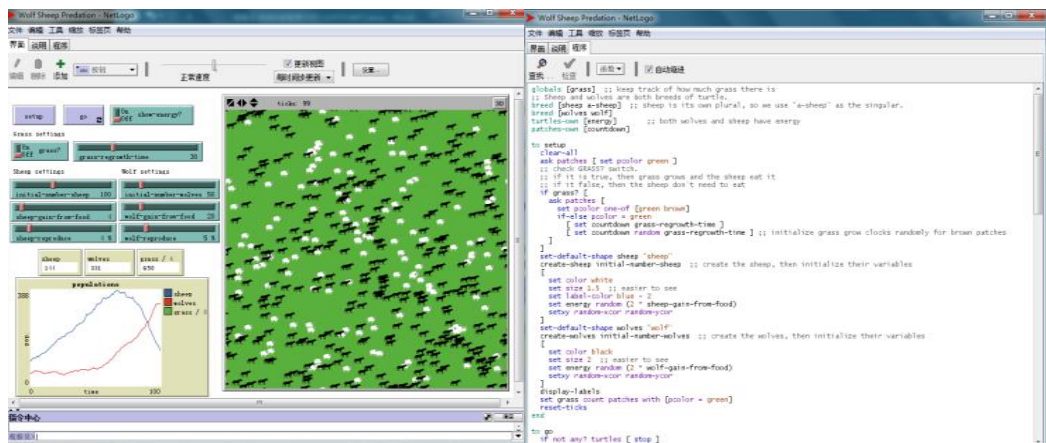
我将这个问题具体化,即考虑一片草原上的狼和羊问题,狼群即为题目中的捕食者,且假设狼群离开羊群是不能够生存的;羊群为食饵,羊群有充足的食物资源(草)来生存。

除此之外,通过查阅资料,我发现不仅两种群有着模型,三种群也有这类似的情况,三种群的情况在我的系统仿真模型中也有一定的体现;还有就是我也考虑了种群的自身阻滞作用。

2. 做题目前的准备工作:

在上课的时候,王老师已经讲过了一些实际模型的建立方法以及求解过程。王老师也花了一部分时间,给我们讲解了 MATLAB 的基本用法以及 MATLAB 的 GUI 工具。自己在上课的时候课下也基本都一一试过这些操作。再加上原先已经对于 MATLAB 有过一定的了解,所以对于此题目入门还是很快的。

说到自己为什么选择这道题目,自己在大二的暑假自己帮读研究生的表姐做过一个项目,简单来说就是“初期城市建设用地的规划”问题。使用的是 NetLogo 软件。当时在了解这个软件的时候,从 NetLogo 它自带的模型库中找到了一个“狼羊草”问题的模型,自己也是因为这个模型,才将 Netlogo 入了门,并且帮助表姐做了一些东西。这个软件的“狼羊草”模型库的界面如下:



图一 Netlogo 软件界面

这个软件和 MATLAB 有异曲同工之处。所以我选题目的时候就毫不犹豫的选择了第十二题：弱肉强食问题--Volterra 模型。为了方便理解，自己也将捕食者和食饵分别看成了狼群以及羊群，其本质上是完全一样的。

3. 系统中只考虑羊群独立生长（食饵独立生长）：

首先考虑最简单情况，就是系统中只考虑羊群独立生长（食饵独立生长）的情况。建立微分方程：

$$\dot{x}_1 = x_1 \times r_1$$

r_1 为羊群（食饵）独立存在时的增长率； x_1 为羊群的实时数量。

4. 系统中存在捕食者（狼群），但是其数量不变化：

再考虑系统中存在捕食者（狼群），但是其数量不变化情况，建立微分方程：

$$\dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2$$

r_1 为羊群（食饵）独立存在时的增长率； λ_1 为捕食者（狼群）掠取食饵的能力； x_1 为羊群的实时数量； x_2 为狼群的实时数量。

5. 系统中存在捕食者（狼群），且其数量随食饵（羊群）变化：

考虑系统中存在捕食者（狼群），且其数量随食饵（羊群）变化，建立微分方程组：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 \end{cases}$$

上面的方程组所代表的的数学模型，就是 Volterra 模型。

其中， r_1 为羊群独立存在时的增长率； r_2 为狼群独自存在时的死亡率； λ_1 为狼群掠取羊群的能力； λ_2 羊群对狼群的供养能力； x_1 为羊群的实时数量； x_2 为狼群的实时数量。

6. 系统在 5 的前提下，考虑种群的自身阻滞作用：

考虑系统中种群的自身阻滞作用，即狼、羊种群数量越多生长速度越慢，这个符合自然地实际情况，建立微分方程组：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 - x_1 \times x_1 \times r_1 \div N_1 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 - x_2 \times x_2 \times r_2 \div N_2 \end{cases}$$

上面的方程组所代表的的数学模型，就是 Logistic-Volterra 模型。

其中， r_1 为羊群独立存在时的增长率； r_2 为狼群独自存在时的死亡率； λ_1 为狼群掠取羊群的能力； λ_2 羊群对狼群的供养能力； x_1 为羊群的实时数量； x_2 为狼群的实时数量； N_1 为羊群的最大数量； N_2 为狼群的最大数量。

7. 系统在 6 的前提下，考虑人工捕获：

考虑系统中存在人工捕获的情况，建立微分方程组：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 - x_1 \times x_1 \times r_1 \div N_1 - \lambda_3 \times x_1 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 - x_2 \times x_2 \times r_2 \div N_2 - \lambda_4 \times x_2 \end{cases}$$

r_1 为羊群独立存在时的增长率； r_2 为狼群独自存在时的死亡率； λ_1 为狼群掠取羊群的能力； λ_2 羊群对狼群的供养能力； x_1 为羊群的实时数量； x_2 为狼群的实时数量； λ_3 为人工捕获羊群的能力； λ_4 为人工捕获狼群的能力； N_1 为羊群的最大数量； N_2 为狼群的最大数量。

8. 再考虑羊的食物——草，对于狼和羊的数量的影响：

考虑羊的食物——草，对于狼和羊的数量的影响，建立微分方程组：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 - x_1 \times x_1 \times r_1 \div N_1 - \lambda_3 \times x_1 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 - x_2 \times x_2 \times r_2 \div N_2 - \lambda_4 \times x_2 \\ \dot{x}_3 = x_3 \times r_3 - x_1 \times \lambda_5 \times x_3 \end{cases}$$

r_1 为羊群独立存在时的增长率； r_2 为狼群独自存在时的死亡率； r_3 为草的增长率； λ_1 为狼群掠取羊群的能力； λ_2 羊群对狼群的供养能力； λ_3 为人工捕获羊群的能力； λ_4 为人工捕获狼群的能力； λ_5 为草对羊群的供养能力； x_1 为羊群的实时数量； x_2 为狼群的实时数量； x_3 为草的实时数量； N_1 为羊群的最大数量； N_2 为狼群的最大数量。

9. 考虑 Volterra 模型下的系统稳定性情况：

考虑只有狼和羊情况下的系统稳定性情况，通过计算可以得到系统的几组稳定性参数。其对于仿真模型正确性的验证是至关重要的。

下面对于下面的非线性微分方程组做稳定性分析。

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 \end{cases}$$

上面的方程组所代表的的数学模型，就是 Volterra 模型。

其中， r_1 为羊群独立存在时的增长率； r_2 为狼群独自存在时的死亡率； λ_1 为狼群掠取羊群的能力； λ_2 羊群对狼群的供养能力； x_1 为羊群的实时数量； x_2 为狼群的实时数量。

首先求平衡点，令 $\dot{x}_1 = 0$ ， $\dot{x}_2 = 0$ ，得 $(0, 0)$ 和 $(\frac{r_2}{\lambda_2}, \frac{r_1}{\lambda_1})$ 。若只考虑 $(\frac{r_2}{\lambda_2}, \frac{r_1}{\lambda_1})$ 的稳定性，则可作变量代换使原点为平稳点，令

$$x_1(t) = \frac{r_2}{\lambda_2} + \rho(t), x_2(t) = \frac{r_1}{\lambda_1} + \varphi(t)$$

带入上面的微分方程组得

$$\begin{cases} \rho'(t) = -\lambda_1 \varphi (\frac{r_2}{\lambda_2} + \rho) \\ \varphi'(t) = \lambda_2 \rho (\frac{r_1}{\lambda_1} + \varphi) \end{cases}$$

设 $|\rho| \ll \frac{r_2}{\lambda_2}, |\varphi| \ll \frac{r_1}{\lambda_1}$ ，去掉二次项使之线性化，得

$$\begin{cases} \rho'(t) = -\frac{\lambda_1 r_2}{\lambda_2} \varphi \\ \varphi'(t) = \frac{\lambda_2 r_1}{\lambda_1} \rho \end{cases}$$

即

$$\begin{bmatrix} \rho'(t) \\ \varphi'(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\lambda_1 r_2}{\lambda_2} \\ \frac{\lambda_2 r_1}{\lambda_1} & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \varphi \end{bmatrix}$$

可以求得上式系数行列式的特征根为 $\pm \sqrt{r_1 r_2} i$ 。故线性方程组的通解为

$$\begin{cases} \rho(t) = \rho_0 \cos(\sqrt{r_1 r_2} t) \\ \varphi(t) = \varphi_0 \sin(\sqrt{r_1 r_2} t) \end{cases}$$

由此可以看出：

(1) ρ 和 φ 的线性化解，既不增长也不衰减，而是连续振动。这意味着平稳是亚稳定

的，这是一种广义的稳定；在平衡点邻域显示出稳定的有界循环。

(2) 令 $\sqrt{r_1 r_2} T = 2\pi$ ，则振动周期 $T = 2\pi / \sqrt{r_1 r_2}$ 。

(3) 捕食者和食饵的总数的振动相位差为 90° 。

(4) 从解中消去时间 t ，得

$$\left(\frac{\rho}{\rho_0}\right)^2 + \left(\frac{\varphi}{\varphi_0}\right)^2 = 1$$

即相空间的一个封闭椭圆形。

从对非线性微分方程组的直接讨论也可以知道轨线是一族以平衡点为中心越来越扩展的封闭曲线。封闭曲线对应着原线性微分方程组的周期解，记周期为 T ， \bar{x}_1 和 \bar{x}_2 分别为在一个周期内 $x_1(t)$ 和 $x_2(t)$ 的平均值，则

$$\bar{x}_1 = \frac{1}{T} \int_0^T \dot{x}_1(t) dt = \frac{1}{T} \int_0^T \frac{1}{\lambda_2} (r_2 + \frac{\dot{x}_2}{x_2}) dt = \frac{r_2}{\lambda_2}$$

其中， $x_2(t)$ 为周期函数， $x_2(T) = x_2(0)$ 。

$$\text{同理 } \bar{x}_2 = \frac{r_1}{\lambda_1}。$$

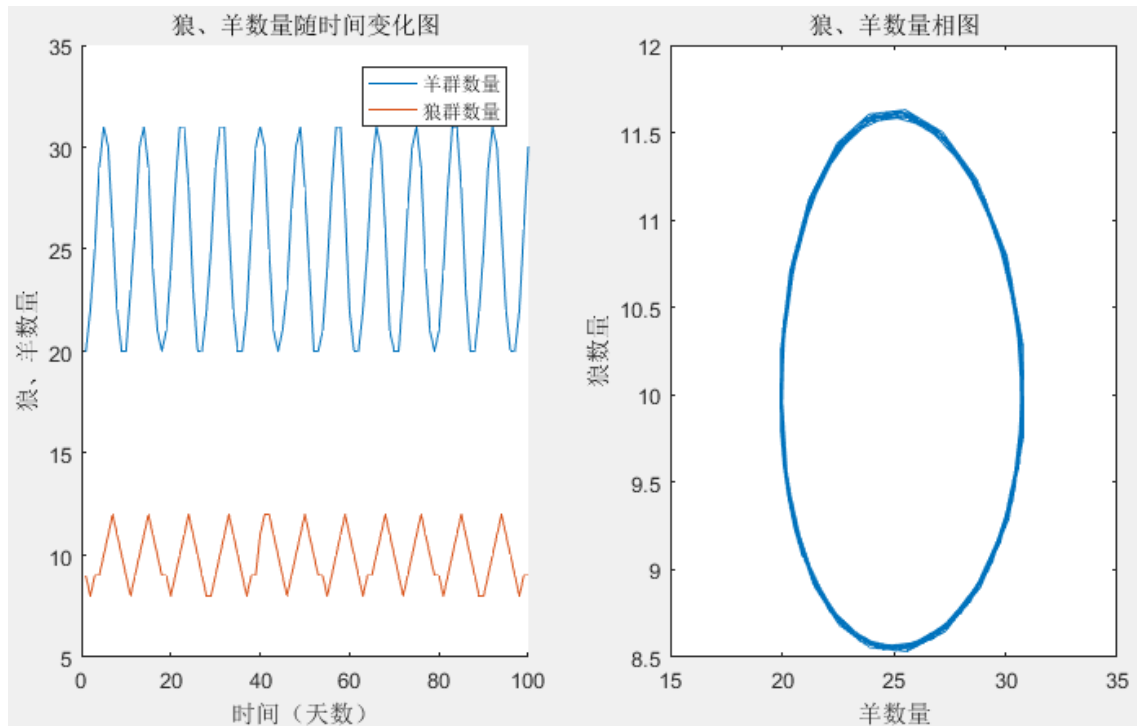
对于周期性变化的 $x_1(t)$ 和 $x_2(t)$ ，用他们的平均值 \bar{x}_1 和 \bar{x}_2 表示其大小。上面的分析表明，羊群的平均数量取决于狼群的方程 $\dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1$ 中的参数 r_2 和 λ_2 ，而狼群的平均数量取决于羊群方程 $\dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2$ 中的参数 r_1 和 λ_1 。当羊的自然增长率 r_1 下降时，狼群的数量将减少，而当狼群掠夺羊群的能力提高时，对于羊群没有影响，只是狼群的数量减少。另一方面，狼群的死亡率 r_2 的上升将导致羊群增多，而羊群对狼群的供养能力 λ_2 的提高，会导致羊群减少。

具体的仿真结果：

1. 在不考虑狼群羊群的自身阻滞作用（即满足 Volterra 模型）时：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 \end{cases}$$

经过选择正确的参数后的仿真图如下：

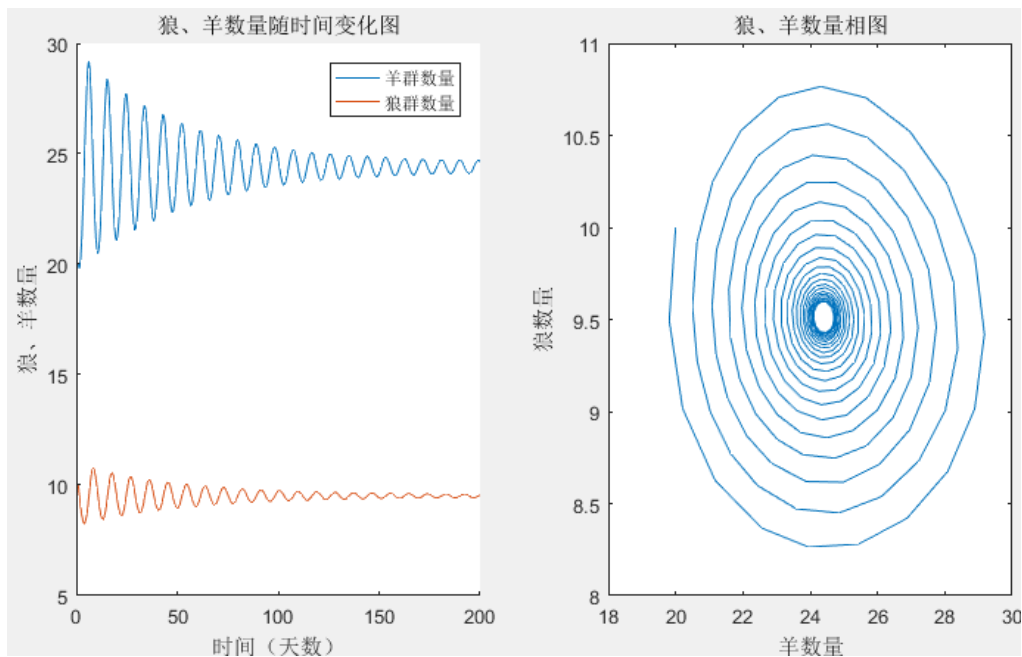


图二 Volterra 模型时域图和相图

2.在考虑狼群羊群的自身阻滞作用（即满足 Logistic-Volterra 模型）时：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 - x_1 \times x_1 \times r_1 \div N_1 - \lambda_3 \times x_1 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 - x_2 \times x_2 \times r_2 \div N_2 - \lambda_4 \times x_2 \end{cases}$$

经过选择正确的参数后的仿真图如下：

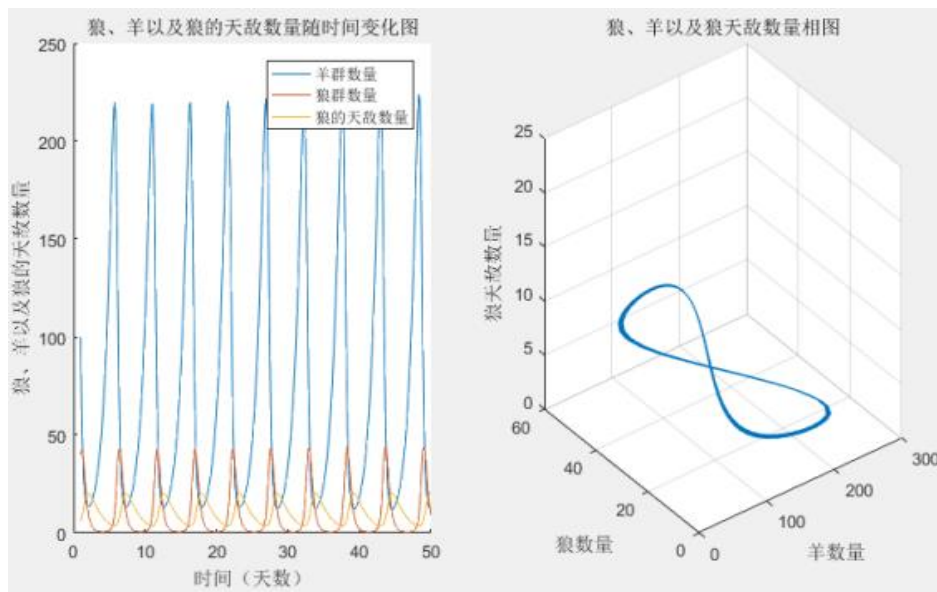


图三 Logistic-Volterra 模型时域图和相图

3.当加上狼群的天敌的时候，考虑三种群的情况：

$$\begin{cases} \dot{x}_1 = x_1 \times r_1 - x_1 \times \lambda_1 \times x_2 - x_1 \times x_1 \times r_1 \div N_1 - \lambda_3 \times x_1 \\ \dot{x}_2 = -x_2 \times r_2 + x_2 \times \lambda_2 \times x_1 - x_2 \times x_2 \times r_2 \div N_2 - \lambda_4 \times x_2 \\ \dot{x}_3 = x_3 \times r_3 - x_1 \times \lambda_5 \times x_3 \end{cases}$$

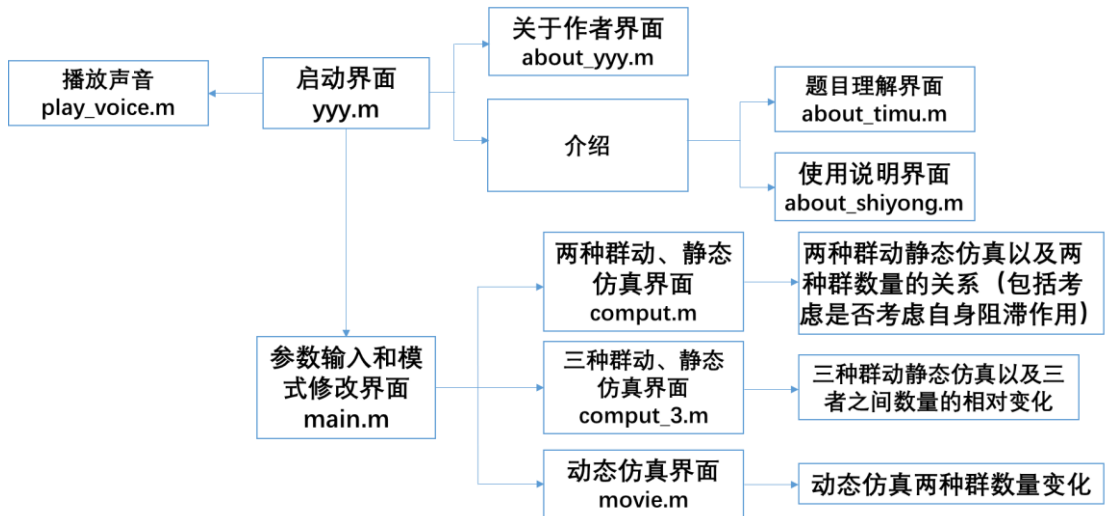
经过选择正确的参数后的仿真图如下：



图四 三种群模型时域图和相图

由仿真可见，可知以上模型基本满足以上稳定性的分析，建立模型基本正确，参数选择基本正确。

三. 系统仿真流程



图五 系统仿真流程图

四. 系统仿真关键点

1. MATLAB 仿真界面的设计

通过 guide 工具进行界面的设计，会自行产生界面和 m 文件，同时生成回调函数。同时也可以用 uicontrol 辅助添加各种界面操作工具。同时对所有位置参数进行归一化处理，适应各种环境。然后使用 set 函数进行界面参数的读取和处理。在实际设计时要注意由 matlab 指令运行延迟产生的 bug，比如界面的关闭。通过建立的模块的 callback 函数，实现仿真的流程。

2. 微分方程组处理

无论是狼群，羊群还是后来所考虑的狼群的天敌，以上三者的数量随时间的变化，均满足微分方程，且均与其他两个种群的数量有关系，故建立一个微分方程组。使用 ode45 函数对于三元微分方程组进行求解，或者是用本身的概念进行求解。例如我使用了两种方法进行求解：

（1）对于 Volterra 模型：

方法一：直接按照定义来求解：

$$\begin{cases} x_1(t) = x_1(t) \times (r_1 + 1) - x_1(t) \times \lambda_1 \times x_2(t) \\ x_2(t) = -x_2(t) \times (r_2 + 1) + x_2(t) \times \lambda_2 \times x_1(t) \end{cases} \quad x_1(0), x_2(0) \text{ 已给出}$$

方法二：使用 MATLAB 自带的 ode45 函数求解（所编写的 MATLAB 代码如下）：

```
%同样也可以用ode45函数，直接对微分方程组求解
odefun = @(t,u) [(u(1) - 11 * u(3)) * u(1) - (u(2) - 13 * u(1)) * u(3)]';
tspan = [1 1];
u0 = [x1_0 x2_0]';
[t1,y] = ode45(odefun,tspan,u0);
```

图六 MATLAB ode45 函数代码图

(2) 对于 Logistic-Volterra 模型：

方法一：直接按照定义来求解：

$$\begin{cases} x_1(t) = x_1(t) \times (r_1 + 1) - x_1(t) \times \lambda_1 \times x_2(t) - x_1(t) \times x_1(t) \times r_1 \div N_1 \\ x_2(t) = -x_2(t) \times (r_2 + 1) + x_2(t) \times \lambda_2 \times x_1(t) - x_2(t) \times x_2(t) \times r_2 \div N_2 \end{cases} \quad x_1(0), x_2(0) \text{ 已给出}$$

方法二：使用 MATLAB 自带的 ode45 函数求解（所编写的 MATLAB 代码如下）：

```
%直接使用ode45求解微分方程组
odefun = @(t,u) [(r1-u(1)+u(1)/N1-u(1)*u(1)) * u(1) - (r2 = 13 * u(1)) * u(3) + u(2) * u(3) / N2]';
tspan = [1 1];
u0 = [x1_0 x2_0]';
[t1,y] = ode45(odefun,tspan,u0);
```

图七 MATLAB ode45 函数代码图

经实际测试，以上两种方法对于两种模型的结果完全一样。

3. 考虑实际情况，对于数据进行合理的约定

1.考虑到实际情况，狼群、羊群以及狼群天敌的数量在任何一个时期都应该为整数值，所以使用 MATLAB 自带的 round 函数对于三个种群的数量分别取整，以符合真实情况。

2.考虑到实际情况，当某一些参数选择不合适时，有可能狼群、羊群的数量会变为负数，这是不符合逻辑的，所以当狼群、羊群数量小于 1 时（有可能为 0-1 之间的小数，也有可能是负数），我都将其数量直接设为 0，以满足实际情况。

3.考虑到实际情况，我也对于仿真的参数进行了一定的约束。

4. 仿真参数的输入

题目要求系统输入包括仿真时间 T ，食饵和捕食者在初始时刻的数量 $x_1(0)$ 和 $x_2(0)$ ，食饵独立生存增长率 r_1 ，捕食者掠取食饵的能力 λ_1 ，捕食者独自存在时死亡率 r_2 ，食饵对捕食者的供养能力 λ_2 。我使用 MATLAB 的 GUI 的可编辑文本实现了这一功能，将每一个可编辑文本中的 Tag 选项变为我们想要的参数，通过使用键盘输入合适的数字，就可以将此数字赋给 Tag 的对应的参数，这样就可以通过键盘输入完成参数的输入。

5. 仿真对象的选择

在 MATLAB 仿真的时候，我设置了三种仿真模式：只对于羊群数量仿真、只对于狼群数量仿真、羊群狼群数量一起仿真。再进行前两种的仿真的时候我使用了 comet 函数进行了动态的仿真，在最后一种仿真（狼群羊群一起仿真时）我使用了时域图和相图共同仿真，使用 subplot 函数将两图一起显示在界面中。

6. 仿真模式的选择

仿真模式有两种选择，一是不考虑种群的自身阻滞作用，二是考虑种群的自身阻滞作用。两者的切换我使用的是 MATLAB GUI 中的单选按钮完成的这一功能，将其中的 Tag 选项变为我们想要的参数，通过使用鼠标选择不选择，就可以将此数字赋给 Tag 的对应的参数，这样就可以通过键盘输入完成参数的输入。再通过 switch 语句对于此参数进行判断，来进行不同的后面的操作。

7. 三种群仿真

为了体现我对于题目的理解，我也编写了三种群的仿真的 MATLAB 代码。但是我并没有做有关输入参数的一些功能，因为这个和前面两种群的情况是重复的，我们需要做的应该更加关注在建立模型方面，所以我直接设定好了参数，这组参数也是通过稳定性检验的参数。也可以进一步反应三种群的动态变化。和两种群的仿真模式的最后一种仿真模式情况一样，我使用了时域图和相图共同仿真，使用 subplot 函数将两图一起显示在界面中。

8. 图形动态仿真

一般狼群和羊群都生活在一片草原中，在一片区域中共同生存，所以我设计了一个动态的仿真模型，即动画，来实时仿真狼群羊群的数量变化以及位置变化。具体实现过程如下：

根据计算出来的每一时刻的狼群数量以及羊群数量，建立随机矩阵，来代表羊群狼群的位置坐标。例如，某一时刻狼群有 50 只狼，我就会使用 rand 函数，生成两个不同的数据都为 1-1000 行数为 1 列数为 50 的随机矩阵，一个随机矩阵代表行的坐标，另一个随机矩阵代

表列的坐标。羊群的话同理，这样就可以实现狼羊“生活”在绿绿草原的情况。在旁边我也会实时显示狼群和羊群的数量变化，其具体的参数也是可以在仿真参数输入界面输入的。

9. 普通二维仿真

在二维仿真中，我也设定了仿真对象以及仿真模式的选择。可以看到不同的对象的图像以及存在自身阻滞作用和不存在自身阻滞作用时的两种群的数量变化情况，也可以从相图中看出两者之间的关系，并从中判断出两种群组成的系统的稳定性。

10. 图像保存

为了方便使用者对于图像的日后处理，我也编写了图像保存的功能，并且将每一次存储的图像以不同的名字命名，我的命名方式为：1. 如果不考虑种群的自身阻滞作用，则保存图像的名称为：don_t_have_self_block+日期_月份_年份+第几张图片。2. 如果考虑种群的自身阻滞作用，则保存图像的名称为：have_self_block+日期_月份_年份+第几张图片。我采取的是自动存储的方式，存到当前.m 文件的路径下。

五. 程序运行指南

1. 关于文件及内容

此弱肉强食问题仿真模拟系统共 9 个 m 文件：about_yyy.m、about_timu.m、about_shiyong.m、play_voice.m、yyy.m、main.m、compute.m、comput_3.m、movie.m。

并有 6 个 fig 文件：about_yyy.fig、main.fig、about_timu.fig、about_shiyong.fig、movie.fig、yyy.fig；

还有 1 张图片和 1 个声音文件：start.jpg、sheep_voice2.wav；

yyy.m 文件：系统仿真入口，运行后进入欢迎界面，其中也有介绍界面以及作者简介界面。

about_yyy.m 文件：欢迎界面的菜单的一个选项，里面是 GUI 生成的作者介绍；

about_timu.m 文件：欢迎界面的菜单的一个选项，里面是 GUI 生成的作者对于题目的理解；

about_shiyongm 文件：欢迎界面的菜单的一个选项，里面是 GUI 生成的仿真系统的使用介绍说明；

play_voice.m 文件：一个音频的播放文件，在环境界面点击“进入仿真”按钮时，会播放出音频文件。

main.m 文件：仿真系统的主界面，包括系统参数设置界面，用于各种参数输入、仿真模式选择、以及选择仿真的对象等；

comput.m 文件：用于计算并仿真两个种群下的结果，包括考虑自身阻滞作用和不考虑自身阻滞作用两种情况，同时也包括数据分析以及动静曲线绘制，以及相图绘制，也加入了图像保存的功能。

comput_3.m 文件：用于仿真三个种群下的结果，使用自己计算的一组参数来显示三种

群的关系，其中包括静态曲线绘制，以及相图绘制。

movie.m 文件：用于动态显示羊狼种群的变化情况，来实时仿真狼群羊群的数量变化以及位置变化。

2. 操作系统介绍

在命令行窗口输入 yyy 开始运行程序，进入欢迎界面，在欢迎界面的左上角，有菜单栏，有介绍和关于作者两个下拉菜单，点击关于作者按钮可以查看作者信息以及参考资料的情况。介绍中有题目理解和使用说明两个按钮。可以点击题目理解按钮来看作者对于题目的理解，点击使用说明按钮来看仿真系统的使用说明。右下角有两个按钮，点击退出按钮可以关闭此界面并退出系统，点击进入仿真按钮后会播放羊叫的音频然后自动进入仿真界面。

进入仿真主界面中，有仿真参数的输入以及仿真对象以及仿真模式等的选择，设置好所有的参数后（参数一定要合理否则会报错，而且不能为空），点击开始仿真按钮开始仿真，默认是在默认的系统参数中，进入不考虑种群自身阻滞的模式进行两种群仿真。也可以选择仿真对象对于羊群数量、狼群数量分别仿真，也可以选择对于羊群和狼群同时仿真。还可以选择是否考虑种群自身阻滞作用的情况。点击开始仿真按钮后，只对于一个种群进行仿真的模式会使用动态仿真的模式，只画出时域图，对于两个种群同时仿真的会出现静态的时域图以及相图，通过相图可以更好地看出种群是否趋于稳定以及趋于稳定的速度变化情况。

三种群仿真时，是系统按照默认的参数以及时间进行仿真以及相图仿真。在羊种群和狼种群同时仿真的时候，系统会自动保存图像在 main.m 的路径下。

动态系统仿真时，是使用在参数设置页面设置好的参数来使用动画来实时仿真狼群羊群的数量变化以及位置变化。

在仿真系统参数输入时，也设置了参数复位按钮，复位到原来系统默认的参数中，来方便用户操作。

六. 系统模型的亮点以及新颖的地方

1. 题目要求中没有考虑两种群的自身阻滞情况，但是我考虑了，而且可以在考虑和不考虑之间进行进一步的选择。
2. 题目要求中没有考虑三种群的模型，我建立的一个三种群的模型并且使用参考资料中的稳定性判据进一步得到了一组可以使三种群稳定的参数，并且通过了仿真的实验验证。
3. 题目中没有要求动态仿真，我建立了一个动态仿真的模型，就像一个“草原”一样，可以实时反映狼羊种群位置以及数量的动态实时数量，以及变化情况。
4. 这个模型的考虑还是比较全面的，例如参数的正确性在仿真之前都已经通过了验证，已经成为合适的参数后，才能进行仿真，否则系统会报错而且会弹出对话框以提醒。还有就是种群的数量一定为正整数，自己在这个方面也进行了约束，当种群数量小于 0 是，强制使其为 0；当种群数量为小数时，使其种群数量化为整数。
5. 在仿真系统中使用了相图，这样可以比时域图更加清晰的反映整个系统的稳定性的情况。
6. 设置了图像的保存功能，方便使用者进一步处理图像来进行进一步的研究。而且可以多次保存且后者不会将前者已存储的图像覆盖。
7. 多界面形式，不是单一界面，所以要考虑不同界面转换的情况，更要考虑参数传递

的正确性、准确性以及稳定性。

8. 最重要的一点，所有代码以及界面全部自己独立完成，绝不存在抄袭的现象！

七. 程序运行实例分析

下面介绍仿真程序运行过程，并截取仿真过程中的部分图片。

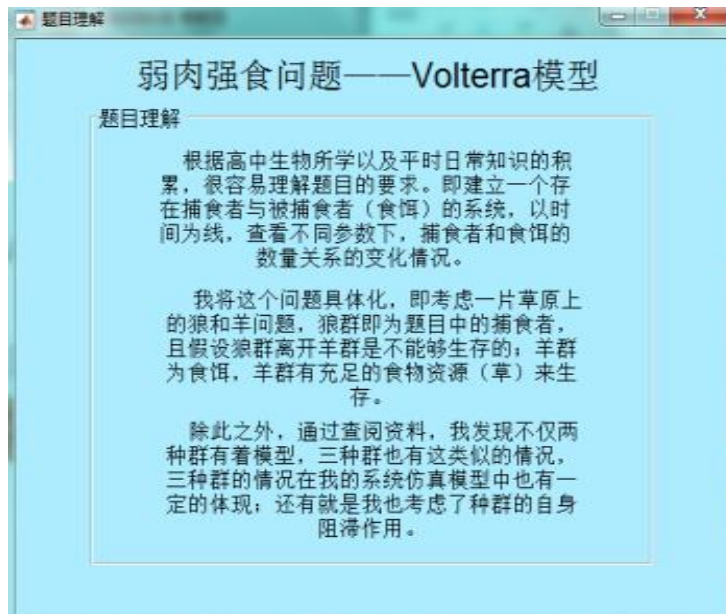
在命令行窗口输入 yyy 开始运行程序，进入欢迎界面。



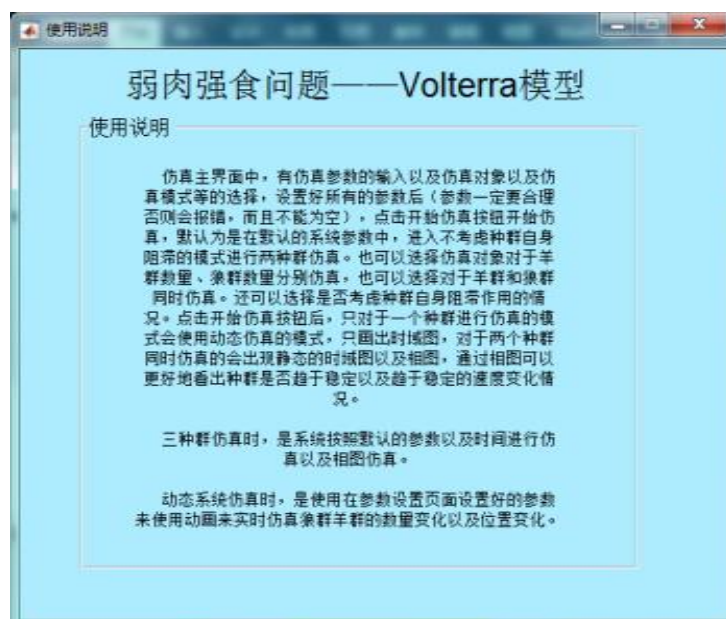
点击左上角的介绍界面，会显示下拉菜单。



点击题目理解，会出现“题目理解”的对话框，里面有作者对于题目的理解。



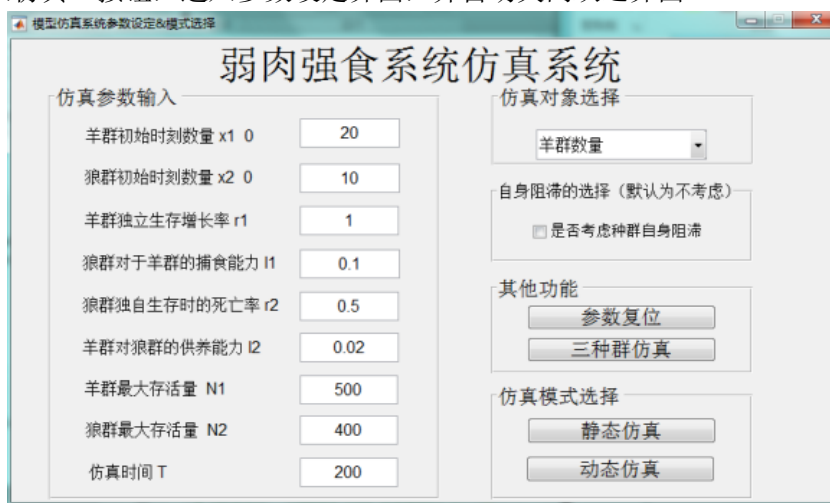
点击使用说明，会出现“使用说明”的对话框，里面有此仿真系统的使用说明。



点击关于作者，会出现“关于作者”的对话框，里面有关于作者的介绍和参考书目的情况。



点击“进入仿真”按钮，进入参数设定界面，并自动关闭欢迎界面



输入合理的参数，选择适宜的仿真对象以及是否考虑种群的自身阻滞作用，如参数不符合要求，则会出现（举一个不符合的例子，这样的约束不止一个）



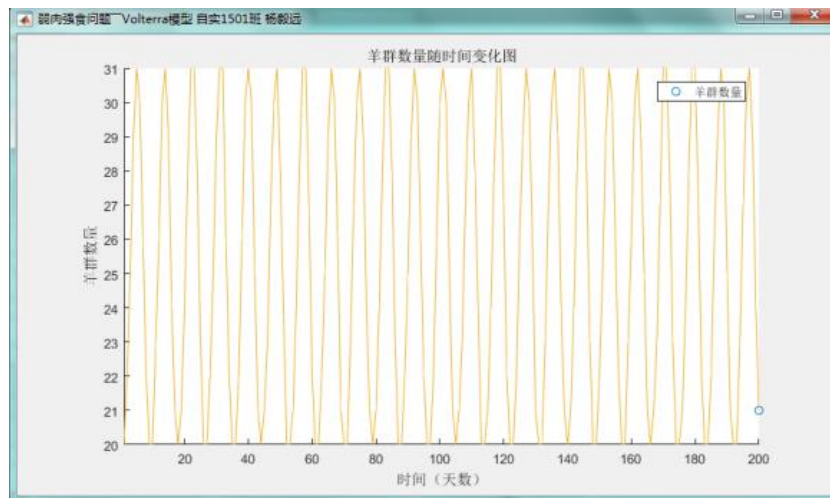
输入合理的仿真时间，系统设定的仿真时间为 10-1000，小于 10 和大于 1000 都会出现报错信息



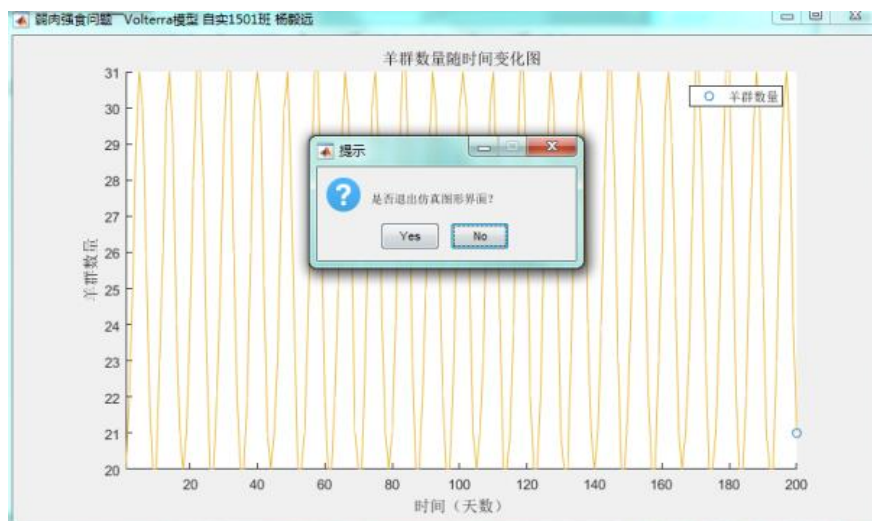
所有的信息都不能空，否则也会报错



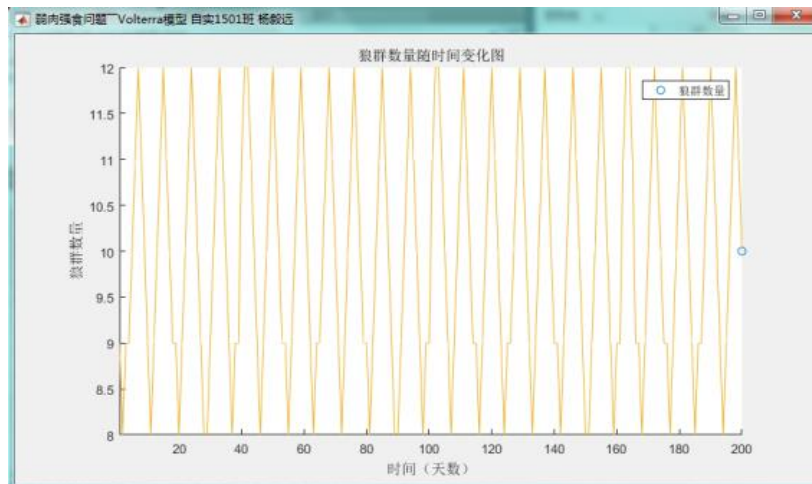
在正确的参数输入的情况下（假设为默认参数），仿真对象选择为羊群数量，点击开始仿真按钮，弹出波形对话框



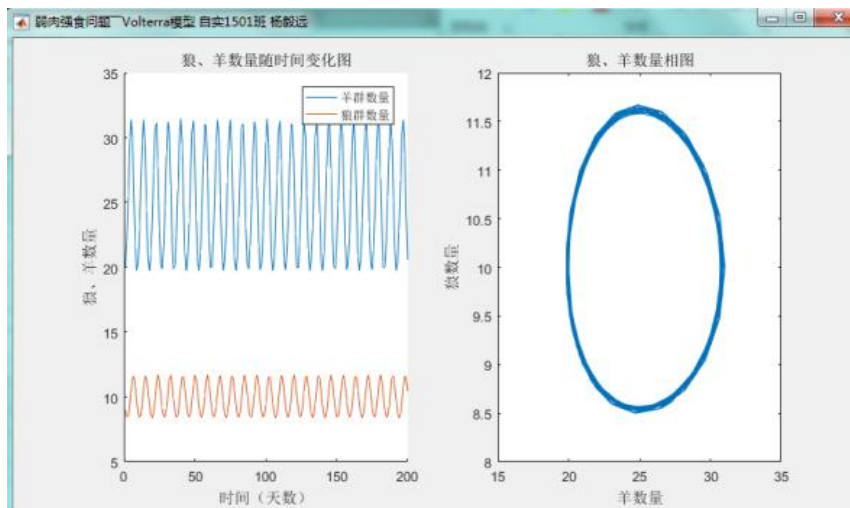
关闭时会弹出对话框，是否确定退出



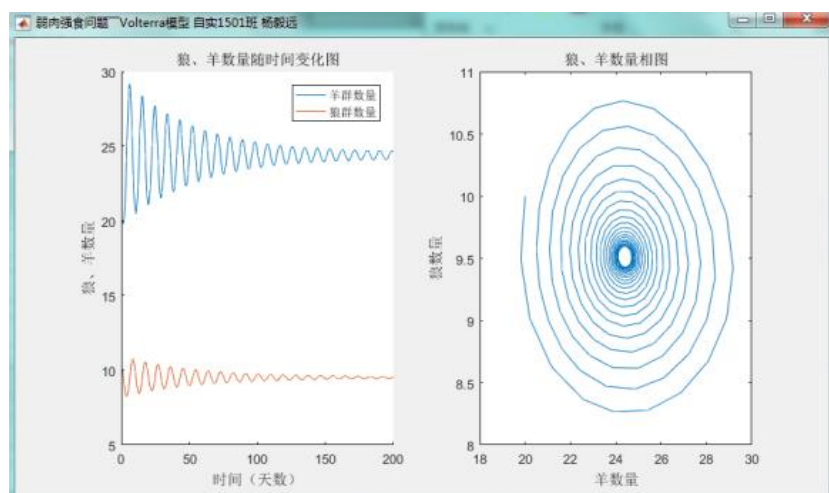
在正确的参数输入的情况下（假设为默认参数），仿真对象选择为狼群数量，点击开始仿真按钮，弹出波形对话框



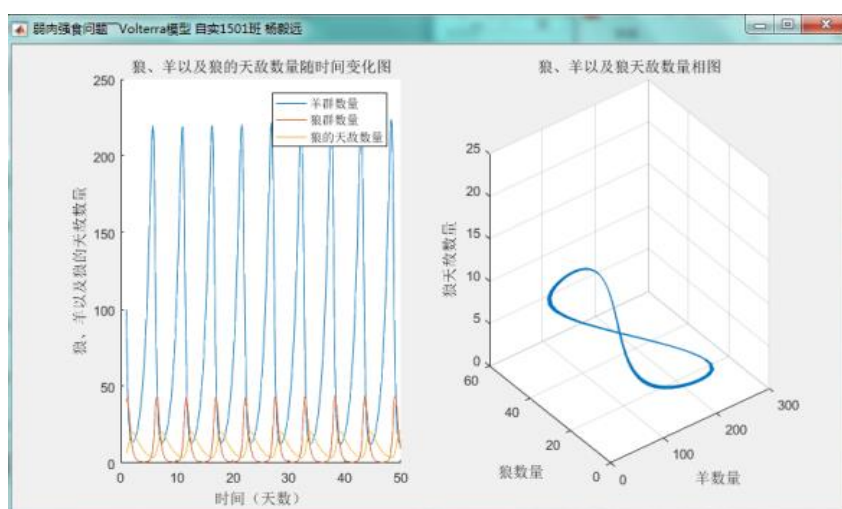
在正确的参数输入的情况下（假设为默认参数），仿真对象选择为狼羊数量，点击开始仿真按钮，弹出波形对话框（波形会自动保存到默认的文件夹中，名称为 don_t_have_self_block+日期_月份_年份+第几张图片.jpg）



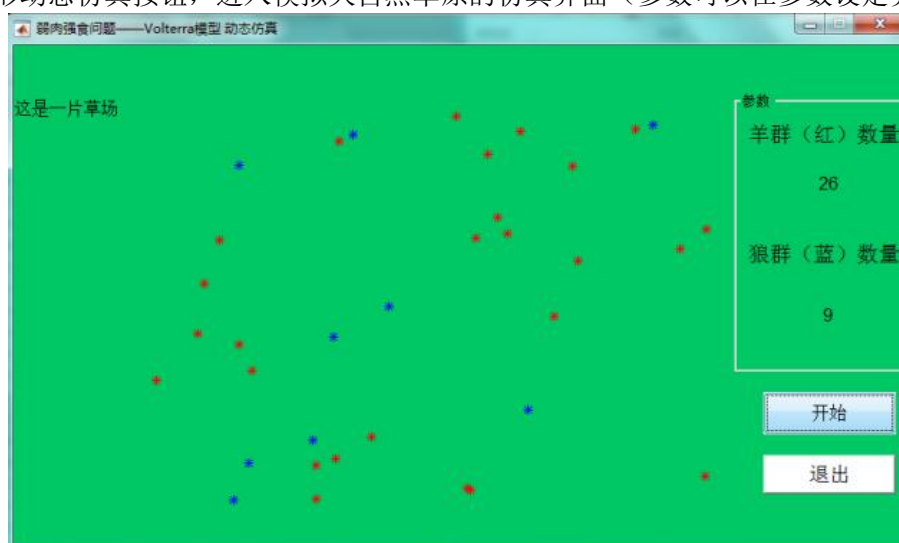
以上都是不考虑种群自身阻滞作用的情况，如考虑种群自身阻滞作用时，羊群数量和狼群数量是类似的，下面只看狼羊数量，仿真对象选择狼羊对象，在是否考虑种群自身阻滞的单选框中打上对勾，点击开始仿真按钮，会弹出对话框（默认参数情况下）（波形会自动保存到默认的文件夹中，名称为 have_self_block+日期_月份_年份+第几张图片.jpg）



点击三种群仿真按钮，显示的是狼群、羊群以及狼的天敌的数量变化情况（默认参数情况下）（没有做有关参数设定的功能，只有一组作者提前算好的数据）



点击图形动态仿真按钮，进入模拟大自然草原的仿真界面（参数可以在参数设定界面设定）



参数仿真界面的参数复位按钮的功能是恢复到系统默认的参数。

八. 心得体会

由于之前已经对于这个题目有过一定的了解,而且上课的时候也跟着老师学习了系统建模以及 MATLAB 的知识,所以这个题目做起来还是比较快的。但是其中的计算稳定性是花了我很多很多的时间,为了找到一组十分合理的参数,我也查阅了相关的书籍,在网上也找到了很多的资料,通过努力自己也成功的解决了问题,并找到了两种群和三种群的稳定点,这为调整参数时,节约了很多很多的时间。

这个课程的本质是让我们使用 MATLAB 来解决一个实际问题,所以建立模型的稳定性、正确性与 MATLAB 的仿真界面同等重要。作为一个毫无艺术细胞的人,自己在界面设计以及系统仿真流程这方面也是下了很大的功夫。尽量做到了结构清晰简洁大方,不刻意追求花里胡哨的外表。将模型和 MATLAB 很好的融合在了一起。

自己在大二的时候和室友参加过一些数模的比赛,为了准备比赛我们也共同学习了 MATLAB,虽然仅仅是入门,但是我们就已经了解了 MATLAB 的强大的功能。

对于 MATLAB 的印象加深是在我大三的暑假刚刚开学的时候,我的教师班主任是周纯杰老师,有一次我们在南一楼开完班会以后,去他的实验室参观,看到有两个他的研究生正在做一个多个水箱排水系统的项目,除了使用软件以及硬件以外,学长也向我们介绍了他们使用 MATLAB 进行仿真,来检验参数的变化,因为这样很直观而且十分方便操作,他们当时也和我说一定要好好掌握 MATLAB。自己也在开始学习这门课程时,更加深入的学写了 MATLAB。做完这个课程设计以后,我对于使用 MATLAB 进行系统的仿真也有了更深一层的理解。

我觉得使用 MATLAB 制作一个仿真系统是一个既简单又复杂的事。简单在于 MATLAB 强大的功能,如此多的库函数已经方便的图形化界面设计。而难是因为首先要进行模型的建立,从一个理想化的模型,到越来越接近实际的一个真实的模型,我还没有达到能够完全抽象一个模型的能力。

在进行这个课程设计的过程中,我对于 MATLAB 图形化界面的设计和分析问题解决问题的能力已经有了一定的进步。也已经可以使用 MATLAB 的 GUI 功能搭建基于图形化界面的小系统。

但是这并不是终点, MATLAB 还有更多可以学习的地方等着我去学习,对于系统模型的建立以及建立后的仿真,自己也应该再下功夫去深究。自己也应该带着学习 MATLAB 的态度以及做本课程设计的态度去进行接下来的工作与学习。

最后还是应该感谢帮助我完成课程设计的老师和学长、学姐和同学们。

九. 参考书籍及资料

1. 《系统建模与仿真》 齐欢 王小平 编著 清华大学出版社
2. 《MATLAB 程序设计与应用 (第二版)》 刘卫国 编著 高等教育出版社
3. 《MATLAB 实用教程》 苏金明 阮沈勇 编著 电子工业出版社
4. 《确定 Lotka-Volterra 生态系统模型高精度参数的研究》 唐静波 胡智渊 张彦琼
5. 《基于 MATLAB 的三种群食饵——捕食者模型数值解》 山玉林 唐家德
6. 《NetLogo_manual_chinese》

十. 部分代码附录(手写, 非 GUI 生成)

1. main.m

```
%建立对话框标题
set(handles.figure1, 'name', '模型仿真系统参数设定&模式选择');
%set(handles.figure1, 'color', [0, 200, 100]/255);
set(handles.figure1, 'units', 'normalized');
set(handles.figure1, 'position', [0.2, 0.2, 0.6, 0.6]);

function run_Callback(hObject, eventdata, handles)
%定义变量
global x1_0 x2_0 r1 l1 r2 l2 T N N1 N2 i1

%判断单选按钮的返回值, 并赋给 N 做以后的处理
if get(handles.N, 'Value')==1
    N = 1;
else
    N = 0;
end
i1 = i1 + 1; %记录保存的是第几张图片
%将输入的字符转化为数字, 并赋给变量做以后的处理
choose = get(handles.choose, 'value');
x1_0 = str2double( get(handles.x1_0, 'String'));
x2_0 = str2double( get(handles.x2_0, 'String'));
r1 = str2double( get(handles.r1, 'String'));
l1 = str2double( get(handles.l1, 'String'));
r2 = str2double( get(handles.r2, 'String'));
l2 = str2double( get(handles.l2, 'String'));
T = str2double( get(handles.T, 'String'));
N1 = str2double( get(handles.N1, 'String'));
N2 = str2double( get(handles.N2, 'String'));

%判断输入参数是否为空
if (isnan(x1_0) || isnan(x2_0) || isnan(r1) || isnan(l1) || isnan(r2) || isnan(l2) || isnan(T) || isnan(N1) || isnan(N2))
    msgbox('参数不能为空', 'Error', 'error'); %显示为空的提示对话框
%判断仿真时间是否合理
elseif (T > 1000 || T < 10)
    msgbox('仿真时间请设定在 10-1000 之间', 'Error', 'error'); %不合理的提示对话框
elseif (l1 > 1)
    msgbox('狼群对于羊群的捕食能力没有那么高, 应该小于 1', 'Error', 'error');
elseif (l2 > 1)
```

```

        msgbox('羊群对狼群的供养能力没有那么高，应该小于 1', 'Error','error');
else
    comput(choose,x1_0,x2_0,r1,l1,r2,l2,T,N,N1,N2,i1); %没问题就执行 comput 函数
end

```

```

function reset_Callback(hObject, eventdata, handles)

```

```

%参数重设

```

```

set(handles.x1_0, 'String', '20');
set(handles.x2_0, 'String', '10');
set(handles.r1, 'String', '1');
set(handles.l1, 'String', '0.1');
set(handles.r2, 'String', '0.5');
set(handles.l2, 'String', '0.02');
set(handles.T, 'String', '200');
set(handles.N1, 'String', '500');
set(handles.N2, 'String', '400');

```

```

function pushbutton4_Callback(hObject, eventdata, handles)

```

```

%定义变量

```

```

global x1_0 x2_0 r1 l1 r2 l2 T

```

```

%将输入的字符转化为数字，并赋给变量做以后的处理

```

```

x1_0 = str2double( get( handles.x1_0, 'String'));
x2_0 = str2double( get( handles.x2_0, 'String'));
r1 = str2double( get( handles.r1, 'String'));
l1 = str2double( get( handles.l1, 'String'));
r2 = str2double( get( handles.r2, 'String'));
l2 = str2double( get( handles.l2, 'String'));
T = str2double( get( handles.T, 'String'));

```

```

%判断输入参数是否为空

```

```

if (isnan(x1_0) || isnan(x2_0) || isnan(r1) || isnan(l1) || isnan(r2) || isnan(l2) || isnan(T) )

```

```

    msgbox('参数不能为空', 'Error','error'); %显示为空的提示对话框

```

```

%判断仿真时间是否合理

```

```

elseif (T > 1000 || T < 50 )

```

```

    msgbox('仿真时间请设定在 50-1000 之间', 'Error','error'); %不合理的提示对话框

```

```

else

```

```

    movie(x1_0,x2_0,r1,l1,r2,l2,T); %没问题就执行 movie 函数

```

```

end

```

```

function start_3_Callback(hObject, eventdata, handles)

```

```

comput_3(); %执行三种群仿真函数

```

2. yyy.m

```

%设置对话框内容及显示大小等
set(handles.figure1, 'name', '弱肉强食问题——Volterra 模型 自实 1501 班 杨毅远');
set(handles.figure1, 'units', 'normalized');
set(handles.figure1, 'position', [0.2, 0.2, 0.6, 0.6]);

%显示背景图片，全屏，在背景的最后
ha = axes('units','normalized','pos',[0 0 1 1]);
uistack(ha,'down');
img = imread('start.jpg');
image(img);
colormap gray
set(ha,'handlevisibility','off','visible','off');
axis off

%登录按钮句柄的创建
uicontrol('style', 'push',...
    'units', 'normalized',...
    'position', [0.75, 0.2, 0.2, 0.1],...
    'string', '进入仿真',...
    'callback','play_voice();close(gcf);main();',...
    'backgroundColor', [45, 126, 0]/255,...
    'fontSize', 16,...
    'ForegroundColor', 'w');

uicontrol('style', 'push',...
    'units', 'normalized',...
    'position', [0.75, 0.05, 0.2, 0.1],...
    'string', '退出系统',...
    'callback','[close(gcf)]',...
    'backgroundColor', [1, 1, 2]/255,...
    'fontSize', 16,...
    'ForegroundColor', 'w');

function timulijie_Callback(hObject, eventdata, handles)
about_timu(); %题目理解界面

function shiyongshuoming_Callback(hObject, eventdata, handles)
about_shiyong(); %使用说明界面

function guanyuzuoazhe_Callback(hObject, eventdata, handles)
about_yyy(); %作者介绍界面

```

3. compute.m

```

function[] = comput(choose,x1_0,x2_0,r1,l1,r2,l2,T,N,N1,N2,i1)

```

```

%假定此问题为狼羊问题，即食饵是羊，捕食者的狼

%choose 是仿真的选择情况 choose = 1 为只仿真羊数量变化
%                               choose = 2 为只仿真狼数量变化
%                               choose = 3 为同时仿真狼羊数量变化

%N 是是否考虑种群自身阻滞作用的情况 0 为不考虑
%                               1 为考虑
%T 是仿真时间
%x1_0 是羊的初始数量
%x2_0 是狼的初始数量
%r1 羊群独立生存的增长率
%l1 狼群对于羊群的捕食能力
%r2 狼群独自生存时的死亡率
%l2 羊群对狼群的供养能力
%N1 羊群最大存活量
%N2 狼群最大存活量

%设置标题的属性
H1 = figure('color', [1,1,1], ...
            'units', 'normalized', ...
            'position', [0.2, 0.2, 0.6, 0.6], ...
            'Name', '弱肉强食问题——Volterra 模型 自实 1501 班 杨毅远', ...
            'NumberTitle', 'off', ...
            'MenuBar', 'none', ...
            'CloseRequestFcn', @my_closereq);
set(H1, 'color', [240 240 240]/255);
%清空之前所有的显示
cla

%判断是否考虑种群自身阻滞作用的情况
switch N
case 0 %0 为不考虑
    x1 = x1_0;
    x2 = x2_0;
    result_x1 = zeros(1,T); %建立两个 1 行 T 列的空矩阵 result_x1&result_x2
    result_x2 = zeros(1,T);
    for t = 1 : T %模型求解
        x1 = (x1 * (1 + r1)) - (x1 * l1 * x2 );
        x2 = (x2 * (1 - r2)) + (x2 * l2 * x1 );
        if x1 < 0 %根据实际情况，出现负数，立刻归 0
            x1 = 0;
        end
    end

```

```

if x2 < 0
    x2 = 0;
end
%根据实际情况 种群数量一定为正整数，故使用 round 函数进行近似取整
result_x1(1,t) = x1 ; %x1 为食饵最后结果
result_x2(1,t) = x2 ; %x2 为捕食者最后结果
end

```

```

%同样也可以用 ode45 函数，直接对微分方程组求解
odefun = @(t1,x)[(r1 - l1 * x(2)) * x(1) (- (r2 - l2 * x(1)) * x(2))];
tspan = [1 T];
x0 = [x1_0 x2_0]';
[t1,y] = ode45(odefun,tspan,x0);

```

```

switch choose    %choose 是仿真的选择情况
case 1          %choose = 1 为只仿真羊数量变化
    cla        %清除所有之前的显示

    comet(round(result_x1));    %动态画图
    xlabel('时间（天数）');
    ylabel('羊群数量');
    title('羊群数量随时间变化图');    %添加图像标题
    legend('羊群数量');    %加注释
case 2          %choose = 为只仿真狼数量变化
    cla

    comet(round(result_x2));
    xlabel('时间（天数）');
    ylabel('狼群数量');
    title('狼群数量随时间变化图');    %添加图像标题
    legend('狼群数量');
case 3          %choose = 3 为同时仿真狼羊数量变化
    cla
    subplot(121);    %分为左右两个界面，界面一显示随时间的变化
    hold on
    plot(result_x1);
    plot(result_x2);
    xlabel('时间（天数）');
    ylabel('狼、羊数量');
    title('狼、羊数量随时间变化图');    %添加图像标题
    legend('羊群数量','狼群数量');

    subplot(122);    %界面二显示，两者之间的关系

```

```

        plot(y(:,1),y(:,2));
        xlabel('羊数量');
        ylabel('狼数量');
        title('狼、羊数量相图');    %添加图像标题
        hold off

        name1 = strcat('don_t_have_self_block ',date);
        name2 = strcat(name1,'_');
        name = strcat(name2,num2str(i1));    % 保存 图 像 的 名 称 为 :
don_t_have_self_block+时间+第几张
        saveas(gcf,name,'jpg'); %保存图像
    end

case 1    %考虑自身阻滞作用
    x1 = x1_0;
    x2 = x2_0;

    %直接使用 ode45 求解微分方程组
    odefun = @(t1,x)[(r1-x(1)*r1/N1-l1*x(2))*x(1) -(r2 - l2 * x(1)) * x(2)) + r2 * x(2) *
x(2)/N2];
    tspan = [1 T];
    x0 = [x1_0 x2_0];
    [t1,y] = ode45(odefun,tspan,x0);

    %与上面类似
    switch choose
        case 1
            cla
            plot(t1,y(:,1));

            xlabel('时间（天数） ');
            ylabel('羊群数量');
            title('羊群数量随时间变化图');    %添加图像标题
            legend('羊群数量');
        case 2
            cla
            plot(t1,y(:,2));

            xlabel('时间（天数） ');
            ylabel('狼群数量');
            title('狼群数量随时间变化图');    %添加图像标题
            legend('狼群数量');
        case 3
            cla

```

```

subplot(121);
hold on
plot(t1,y(:,1));
plot(t1,y(:,2));
xlabel('时间（天数）');
ylabel('狼、羊数量');
title('狼、羊数量随时间变化图');    %添加图像标题
legend('羊群数量','狼群数量');

subplot(122);
plot(y(:,1),y(:,2));
xlabel('羊数量');
ylabel('狼数量');
title('狼、羊数量相图');    %添加图像标题
hold off

name1 = strcat('have_self_block ',date);
name2 = strcat(name1,'_');
name = strcat(name2,num2str(i1)); %保存图像的名称为: have_self_block+
时间+第几张

saveas(gcf,name,'jpg'); %保存图像

end

end

end

%退出对话框的建立
function my_closereq(src, callbackdata)
% Close request function
% to display a question dialog box
selection = questdlg('是否退出仿真图形界面? ', ...
    '提示', ...
    'Yes','No','No');
switch selection %选择是否退出
    case 'Yes'
        delete(gcf);
    case 'No'
        return
end
end
end

```

4. compute_3.m

```
function[] = comput_3()
```

%假定此问题为狼羊以及狼的天敌的问题，即食饵是羊，捕食者的狼，再加上考虑的狼的天

敌的存在

%设置标题的属性

```
H1 = figure('color', [1,1,1], ...  
            'units', 'normalized', ...  
            'position', [0.2, 0.2, 0.6, 0.6], ...  
            'Name', '弱肉强食问题——Volterra 模型 自实 1501 班 杨毅远', ...  
            'NumberTitle', 'off', ...  
            'MenuBar', 'none', ...  
            'CloseRequestFcn', @my3_closereq);  
set(H1, 'color', [240 240 240]/255);  
%清空之前所有的显示  
cla
```

```
x1_0 = 100; %x1_0 是羊的初始数量  
x2_0 = 40; %x2_0 是狼的初始数量  
x3_0 = 6; %x3_0 是狼的天敌初始数量  
u1 = 0.1; %狼的天敌对于狼的捕食能力  
l1 = 0.1; %l1 狼群对于羊群的捕食能力  
l2 = 0.02; %l2 羊群对狼群的供养能力  
l3 = 0.06; %l3 狼群对狼的天敌的供养能力  
r1 = 1; %r1 羊群独立生存的增长率  
r2 = 0.5; %r2 狼群独自生存时的死亡率  
r3 = 0.6; %r3 狼的天敌独自生存时的死亡率  
T = 50; %T 是仿真时间
```

%用 ode45 函数，直接对微分方程组求解

```
odefun = @(t1,x)[(r1 - l1 * x(2)) * x(1) (-r2 - l2 * x(1) + u1 * x(3)) * x(2) x(3)*(-r3 + l3 * x(2))];  
tspan = [1 T];  
x0 = [x1_0 x2_0 x3_0];  
[t1,y] = ode45(odefun,tspan,x0);
```

cla

%分为左右两个界面，界面一显示随时间的变化

```
subplot(121);  
hold on  
plot(t1,y(:,1));  
plot(t1,y(:,2));  
plot(t1,y(:,3));  
xlabel('时间（天数）');  
ylabel('狼、羊以及狼的天敌数量');  
title('狼、羊以及狼的天敌数量随时间变化图'); %添加图像标题  
legend('羊群数量','狼群数量','狼的天敌数量');
```

```

%界面二显示，三者之间的关系
subplot(122);
plot3(y(:,1),y(:,2),y(:,3));
xlabel('羊数量');
ylabel('狼数量');
zlabel('狼天敌数量')
title('狼、羊以及狼天敌数量相图');    %添加图像标题
grid on
hold off

end

%退出对话框的建立
function my3_closereq(src, callbackdata)
% Close request function
% to display a question dialog box
selection = questdlg('是否退出仿真图形界面? ', ...
                    '提示', ...
                    'Yes','No','No');
switch selection %选择是否退出
    case 'Yes'
        delete(gcf);
    case 'No'
        return
end
end
end

```

5. movie.m

```

%建立对话框标题
set(handles.figure1, 'name', '弱肉强食问题——Volterra 模型 动态仿真');
set(handles.figure1, 'units', 'normalized');
set(handles.figure1, 'color', [0, 200, 100]/255);
set(handles.figure1, 'position', [0.2, 0.2, 0.6, 0.6]);

uicontrol('style', 'push',...
        'units', 'normalized',...
        'position', [0.843, 0.1, 0.15, 0.08],...
        'string', '退出',...
        'callback','close(gcf);',...
        'backgroundColor', [255, 255, 255]/255,...
        'fontSize', 14,...
        'ForegroundColor', 'k');

function start_Callback(hObject, eventdata, handles)

```

```

%点击开始按钮以后的工作
axes(handles.axes2);

cla;%清空之前坐标轴内残余图像
axis off;%去除坐标轴

global x1_0 x2_0 r1 l1 r2 l2 T ;

x1 = x1_0;
x2 = x2_0;
result_x1 = zeros(1,T); %建立两个 1 行 T 列的空矩阵 result_x1&result_x2
result_x2 = zeros(1,T);

for t = 1 : T/10
    x1 = (x1 * (1 + r1)) - (x1 * l1 * x2 ); %模型求解
    x2 = (x2 * (1 - r2)) + (x2 * l2 * x1 );
    if x1 < 0 %根据实际情况，出现负数，立刻归 0
        x1 = 0;
    end
    if x2 < 0
        x2 = 0;
    end
    %根据实际情况 种群数量一定为正整数，故使用 round 函数进行近似取整
    result_x1(1,t) = round(x1);%x1 为食饵最后结果
    result_x2(1,t) = round(x2);%x2 为捕食者最后结果

    %随机生成一个 1-100 的矩阵 1 行 x1 列，A1 A2 分别作为行列向量，一组 A1 A2
    对应一个点，共有 x1 个点来表示羊种群（随机分布的羊种群）
    A1 = 1+(100-1)*rand(1,round(x1));
    A2 = 1+(100-1)*rand(1,round(x1));

    %随机生成一个 1-100 的矩阵 1 行 x2 列，A3 A4 分别作为行列向量，一组 A3 A4
    对应一个点，共有 x2 个点来表示狼种群（随机分布的狼种群）
    A3 = 1+(100-1)*rand(1,round(x2));
    A4 = 1+(100-1)*rand(1,round(x2));

    %视觉停留，表现种群的动态变化
    pause(0.1);
    %set(handles.axes1);
    set(handles.axes2);

    % axis([0,100,0,100]);
    cla; %清除之前的所有显示

```

```

hold on;
plot(A1(1,:),A2(1:),'*r'); %红*表示 x1 只羊
plot(A3(1,:),A4(1:),'*b'); %蓝*表示 x2 只狼

set(handles.text1,'String',round(x1)); %在旁边输出羊群和狼群的动态结果
set(handles.text2,'String',round(x2));
axis([0,100,0,100]); %将坐标固定位 100*100，美观而已
end

```

```

function pushbutton4_Callback(hObject, eventdata, handles)
global button
button = get(hObject,'Value');

```

6. about_yyy.m

```

set(handles.figure1, 'name', '关于作者');

```

7. about_shiyong.m

```

set(handles.figure1, 'name', '使用说明');

```

8. about_timu.m

```

set(handles.figure1, 'name', '题目理解');

```

9. play_voice.m

```

%播放音频函数
function yyy = play_voice(varargin)
    filename='sheep_voice2.wav'; %读取
    [y,fs] = audioread(filename); %存矩阵
    player = audioplayer(y, fs);
    play(player); %播放
    pause(1); %延时
end

```