东南大学

Matlab 仿真实验报告

**课程名称：　　　　　生物系统建模分析　　　　　　　　**

作业周次：　第 5 周

姓　　名：

学　　号：

试用水印

1，理论计算（迭代步骤）

（理论计算在于原理，故列出基本迭代步骤即可，涉及复杂计算处由仿真体现，仅实现原理）

3.2作业：

$$\begin{cases}\dot{S}=a_1ST-a_2S \\ \dot{T}=a_3T-a_4ST\end{cases}，已知\begin{cases}a_1=0.015\\a_2=0.7\\a_3=0.5\\a_4=0.01\end{cases}，\begin{cases}S(0)=100\\T(0)=100\end{cases}$$

(1)显式Euler法：迭代式 $\begin{cases}y'(t)=f(t,y(t))\\y(t_0)=y_0\end{cases} \rightarrow y_{n+1}=y_n+h\underbrace{f(t_n,y_n)}_{对应\ \dot{S}、\dot{T}}$

则 $\begin{cases}S_{n+1}=S_n+h(a_1S_nT_n-a_2S_n)\\T_{n+1}=T_n+h(a_3T_n-a_4S_nT_n)，\\h=\Delta t=0.1\end{cases}$ （或直接代数字→差分同理）

迭代步：0.1年：$\begin{cases}S_1=S_0+\Delta t(a_1S_0T_0-a_2S_0)=100+0.1(0.015\times100\times100-0.7\times100)=108\\T_1=T_0+\Delta t(a_3T_0-a_4S_0T_0)=100+0.1(0.5\times100-0.01\times100\times100)=95\end{cases}$

0.2年：$\begin{cases}S_2=S_1+\Delta t(a_1S_1T_1-a_2S_1)=115.83 \text{（4舍5入 116）}\\T_2=T_1+h(a_3T_1-a_4S_1T_1)=89.49 \text{（取89）}\end{cases}$

0.3年：同理迭代

……

(2) 4阶R-K法：同理有迭代式 $\begin{cases}S_{n+1}=S_n+\frac{1}{6}\Delta t[k_1+2k_2+2k_3+k_4]\\k_1=\dot{S}(t_n,S_n)\\k_2=\dot{S}(t_n+\frac{1}{2}\Delta t,S_n+\frac{1}{2}\Delta t k_1)\\k_3=\dot{S}(t_n+\frac{1}{2}\Delta t,S_n+\frac{1}{2}\Delta t k_2)\\k_4=\dot{S}(t_n+\Delta t,S_n+\Delta t k_3)\end{cases}$

试用水印

0.1年，即 $s_1 = s_0 + \frac{1}{6}\Delta t[k_1+2k_2+2k_3+k_4]$ ，$t_0=0$

$k_1 = \dot{s}(t_0, s_0) = a_1 s_0 T(t_0) - a_2 s_0 = 80$

$k_2 = \dot{s}(t_0+0.05, s_0+0.05k_1) = \dot{s}(\underbrace{0.05}_{t}, \underbrace{104}_{s}) = 1.56T(0.05)-72.8$

$k_3 = \dot{s}(t_0+0.05, s_0+0.05k_2) = \dot{s}(0.05, 100+0.05k_2)$

$k_4 = \dot{s}(t_0+0.1, s_0+0.1k_3) = \dot{s}(0.1, 100+0.1k_3)$

∵ $\dot{s} = a_1 sT - a_2 s$ 即 $\frac{ds}{dt} = s'(t) = a_1 s(t)T(t) - a_2 s(t)$

同理
$$\begin{cases} T_{n+1} = T_n + \frac{1}{6}\Delta t[k_1+2k_2+2k_3+k_4] \\ k_1 = \dot{T}(t_n, T_n) \\ k_2 = \dot{T}(t_n+\frac{1}{2}\Delta t, T_n+\frac{1}{2}\Delta t k_1) \\ k_3 = \dot{T}(t_n+\frac{1}{2}\Delta t, T_n+\frac{1}{2}\Delta t k_2) \\ k_4 = \dot{T}(t_n+\Delta t, T_n+\Delta t k_3) \end{cases}$$

同样利用 $\dot{T}$ 式代入求 $k_{1,2,3,4}$ 斜率

0.2年，同理

……

试用水印

显式欧拉：

- 显式Euler方法：$y_1 - y_0 = f(t_0, y_0)\Delta t$

$$y_2 - y_1 = f(t_1, y_1)\Delta t$$

$$\cdots\cdots$$

$$y_{k+1} - y_k = f(t_k, y_k)\Delta t$$

```
% 定义微分方程组和参数
A1 = 0.015; A2 = 0.7; A3 = 0.5; A4 = 0.01;
f = @(t, y) [A1*y(1)*y(2)-A2*y(1); A3*y(2)-A4*y(1)*y(2)];
tspan = [0, 50];
y0 = [100; 100];
h = 0.1;%步长
t=0.1:h:50;

% 初始化变量
n = length(tspan(1):h:tspan(2));%迭代步数
S = zeros(n, 1);
T = zeros(n, 1);
%S,T用n行1列的全0初始化向量来接住
S(1) = y0(1);
T(1) = y0(2);
% 迭代计算并存储结果，用欧拉式迭代
for i = 1:(n-1)
    y = [S(i), T(i)] + h*f(t(i), [S(i), T(i)])';
    S(i+1) = y(1);
    T(i+1) = y(2);
end
% 绘制每一步长上的具体值
t = tspan(1):h:tspan(2);
plot(t, S,'-o',t,T,'-o');
legend('Shark', 'Tuna');
xlabel('时间');
ylabel('鱼的数目');
title('显式欧拉法');
```

试用水印

定义了 1 个匿名函数，接受参数 t 和长度为 2 的行向量 y（1, 2 分别表示 S, T），输出为长度为 2 的列向量，即微分方程组左侧导数的向量表示

后续迭代中，使用显式欧拉式迭代，将第 i 步的数值行向量+微分方程组左侧导数列向量的转置，结果行向量中取第 1 个，第 2 个即对应的第 i+1 步的迭代值

```matlab
% 定义参数
A1 = 0.015; A2 = 0.7; A3 = 0.5; A4 = 0.01;
tspan = [0, 50];
h = 0.1;%步长

% 初始化变量
n = length(tspan(1):h:tspan(2));%迭代步数
S = zeros(n, 1);
T = zeros(n, 1);
%S,T用n行1列的全0初始化向量来接住
S(1) = 100;
T(1) = 100;

% 迭代计算并存储结果，用欧拉式迭代
for i = 1:(n-1)
    S(i+1) = S(i) + h*(A1*S(i)*T(i) - A2*S(i));
    T(i+1) = T(i) + h*(A3*T(i) - A4*S(i)*T(i));
end

% 绘制每一步长上的具体值
t = tspan(1):h:tspan(2);
plot(t, S,'-o',t,T,'-o');
legend('Shark', 'Tuna');
xlabel('时间');
ylabel('鱼的数目');
title('显式欧拉法');
```

更加明显的迭代式

试用水印

$$\begin{cases}\frac{dS}{dt}=a_1S(t)T(t)-a_2S(t)\\ \frac{dT}{dt}=a_3T(t)-a_4S(t)T(t)\end{cases}$$

，而 $y'=\frac{dy}{dx}=f(x,y)$

$\begin{cases}S\to y\\ t\to x\end{cases}$ $\begin{cases}T\to y\\ t\to x\end{cases}$

$$\begin{cases}k_1=f(x,y)\\ k_2=f(x+\frac{h}{2},y+\frac{h}{2}k_1)\\ k_3=f(x+\frac{h}{2},y+\frac{h}{2}k_2)\\ k_4=f(x+h,y+hk_3)\end{cases} \longrightarrow \begin{cases}k_1=\frac{dS}{dt}(t,S)\\ k_2=\frac{dS}{dt}(t+\frac{h}{2},S+\frac{h}{2}k_1)\\ k_3=\frac{dS}{dt}(t+\frac{h}{2},S+\frac{h}{2}k_2)\\ k_4=\frac{dS}{dt}(t+h,S+hk_3)\end{cases} + \begin{cases}k_1=\frac{dT}{dt}(t,T)\\ k_2=\frac{dT}{dt}(t+\frac{h}{2},T+\frac{h}{2}k_1)\\ k_3=\frac{dT}{dt}(t+\frac{h}{2},T+\frac{h}{2}k_2)\\ k_4=\frac{dT}{dt}(t+h,T+hk_3)\end{cases}$$

$$f=@(t,y)\rightarrow\underbrace{\begin{bmatrix}\frac{dS}{dt}\\ \frac{dT}{dt}\end{bmatrix}}_{2\times1}$$

t: 1×1

y: $[S(t),T(t)]$ 1×2

4 阶 R-K 法：

Runge-Kutta数值计算方法

MATLAB solver
ode45 (5阶Runge-Kutta)

- 四阶Runge-Kutta方法:$y_{k+1}=y_k+\frac{1}{6}\Delta t[k_1+2k_2+2k_3+k_4]$;
- 其中

$$\begin{cases}k_1=f(t_k,y_k)\\ k_2=f\left(t_k+\frac{1}{2}\Delta t,y_k+\frac{1}{2}\Delta t*k_1\right)\\ k_3=f\left(t_k+\frac{1}{2}\Delta t,y_k+\frac{1}{2}\Delta t*k_2\right)\\ k_4=f(t_k+\Delta t,y_k+\Delta t*k_3)\end{cases}$$

- 四阶R-K法：局部截断误差≈ $O(\Delta t^5)$；全局截断误差≈ $O(\Delta t^4)$;

参考补充材料6.4. The Runge–Kutta Methods

试用水印

4 阶显式 Runge-Kutta 算法　已知 $y' = f(x, y)$, $y(x_0) = y_0$, 求 y.

输入 (INPUT)：区间端点 a, b, 初值 (x_0, y_0), 步长 h, 分割子区间个数 N.

输出 (OUTPUT)：最大节点 x_{N+1} 处的近似值 y_{N+1}；

STEP 1. Set $h = \dfrac{b-a}{N}$, $x = x_0$, $y = y_0$（设定步长和初值）,OUTPUT (x, y).

STEP 2. For $n = 1, 2, \ldots, N$, Do STEPS 3-4.

STEP 3. Set

- $K_1 = f(x, y)$
- $K_2 = f(x + \frac{h}{2}, y + \frac{h}{2}K_1)$
- $K_3 = f(x + \frac{h}{2}, y + \frac{h}{2}K_2)$
- $K_4 = f(x + h, y + hK_3)$

STEP 4. Set $x = x + ih$（计算节点值 x_n）

$y = y + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$（计算函数值 y_n）

STEP 5. OUTPUT (x, y).

STEP 6. STOP.

理论迭代见上草稿纸

```
% 定义微分方程组和参数
A1 = 0.015; A2 = 0.7; A3 = 0.5; A4 = 0.01;
f = @(t, y) [A1*y(1)*y(2)-A2*y(1); A3*y(2)-A4*y(1)*y(2)];
tspan = [0, 50];
y0 = [100; 100];
h = 0.1;%步长
t = 0.1:h:50;

% 初始化变量
n = length(tspan(1):h:tspan(2));%迭代步数
S = zeros(n, 1);
T = zeros(n, 1);
%S,T用n行1列的全0初始化向量来接住
S(1) = y0(1);
T(1) = y0(2);

% 迭代计算并存储结果，用4阶runge-kutta法迭代
for i = 1:(n-1)
% 第一步
    k1 =f(t(i), [S(i), T(i)])';
    % 第二步
    k2= f(t(i) + 0.5*h, [S(i), T(i)] + 0.5*h*k1)';
    % 第三步
    k3= f(t(i) + 0.5*h, [S(i), T(i)] + 0.5*h*k2)';
    % 第四步
    k4= f(t(i) + h, [S(i), T(i)] + h*k3)';
    % 计算下一步的S和T
    S(i+1) = S(i) + h*(1/6)*(k1(1) + 2*k2(1) + 2*k3(1) + k4(1));
    T(i+1) = T(i) + h*(1/6)*(k1(2)+ 2*k2(2) + 2*k3(2) + k4(2));
```

试用水印

```
end

% 绘制每一步长上的具体值
t= tspan(1):h:tspan(2);
plot(t, S,'-o',t,T,'-o');
legend('Shark', 'Tuna');
xlabel('时间');
ylabel('鱼的数目');
title('4阶runge-kutta法');
```

同样定义 1 个匿名函数，输出 1 个 2x1 的导数列向量

具体迭代的话就用 f 统一处理 S,T 的导数的变化了

具体细节即

```
% 迭代计算并存储结果，用4阶runge-kutta法迭代
for i = 1:(n-1)
% 第一步
    k1 = f(t(i), [S(i), T(i)])';
    k2 = f(t(i) + 0.5*h, [S(i)+0.5*h*k1(1), T(i)+0.5*h*k1(2)])';
    k3 = f(t(i) + 0.5*h, [S(i)+0.5*h*k2(1), T(i)+0.5*h*k2(2)])';
    k4 = f(t(i) + h, [S(i)+h*k3(1), T(i)+h*k3(2)])';

    S(i+1) = S(i) + h*(1/6)*(k1(1) + 2*k2(1) + 2*k3(1) + k4(1));
    T(i+1) = T(i) + h*(1/6)*(k1(2) + 2*k2(2) + 2*k3(2) + k4(2));
end
```

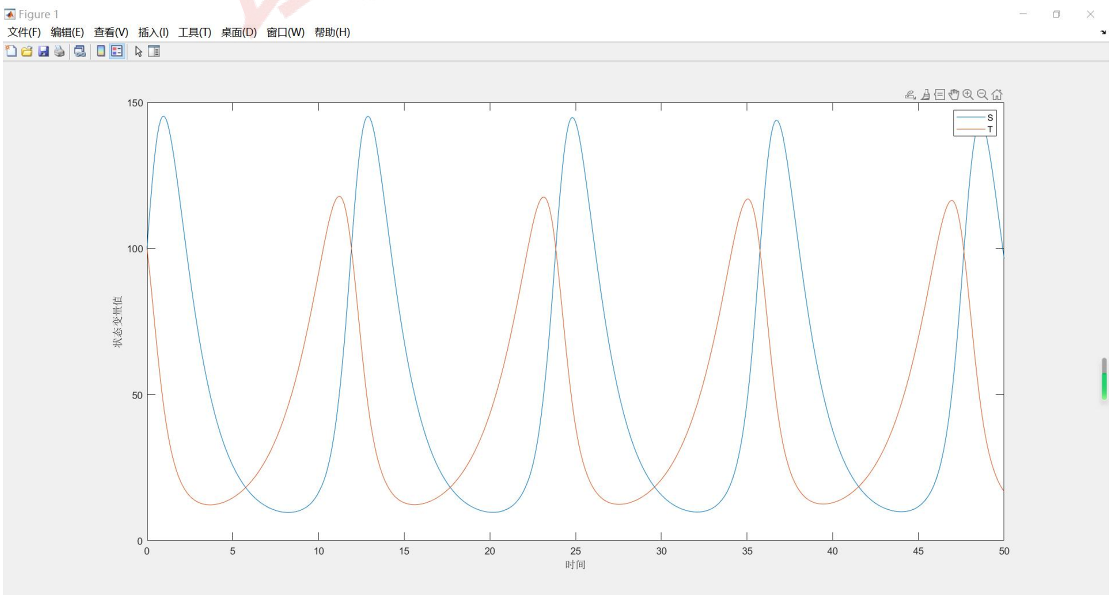
这两块代码结果是一致的

2，结果分析

根据 matlab 仿真+食物链先验知识

下图为 ode45 求解连续（也可以取其他图，欧拉+RK，曲线变化形式一致）

试用水印



刚开始鲨鱼数量增加（捕食金枪鱼），然后金枪鱼数量下降，随之因为食物缺乏所以鲨鱼数量也下降，因为天敌缺少，所以金枪鱼得以繁衍生殖，数量回升，同理食物数量回升，所以鲨鱼数量也增加，如此循环迭代

不过鲨鱼数量是由外界金枪鱼数量反馈影响，不是前馈，所以变化比金枪鱼慢一拍。

结果分析：生物学意义上无非是验证鲨鱼与金枪鱼在一条食物链上，鲨鱼捕食金枪鱼

3，Matlab 源代码（simulink 选做）：代码另附件

Matlab 源代码

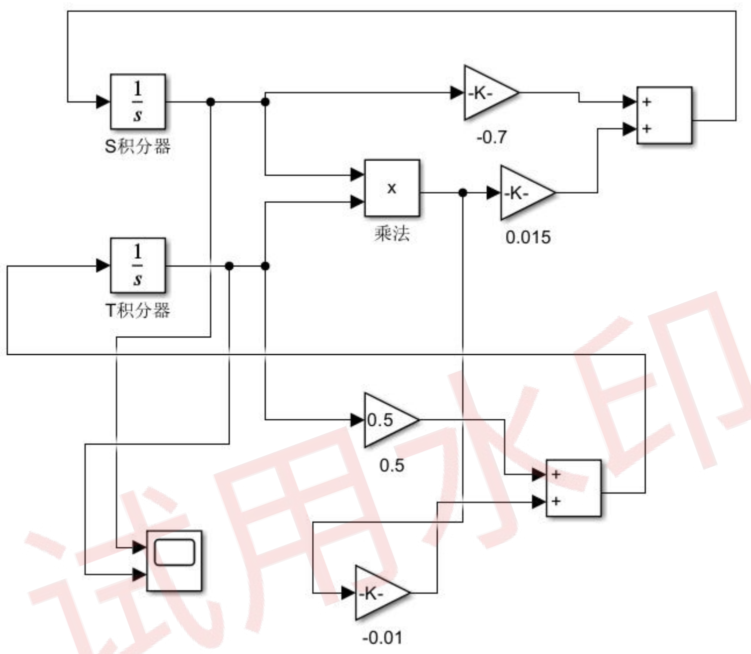
Ode45 见 rk.m，rk1.m，

欧拉法见 rk2.m，rk3.m，

4 阶 R-K 法见 rk4.m，rrk.m，

另外绘图见 RK.fig（ode45），其他方法运行代码同样得图

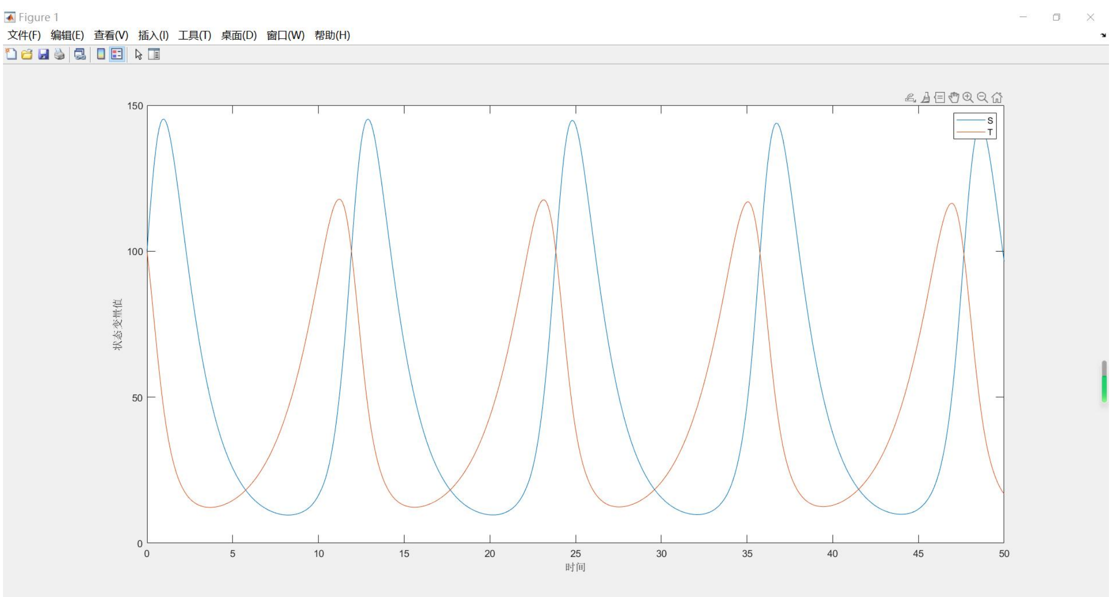
simulink 见附件 sim1.slx

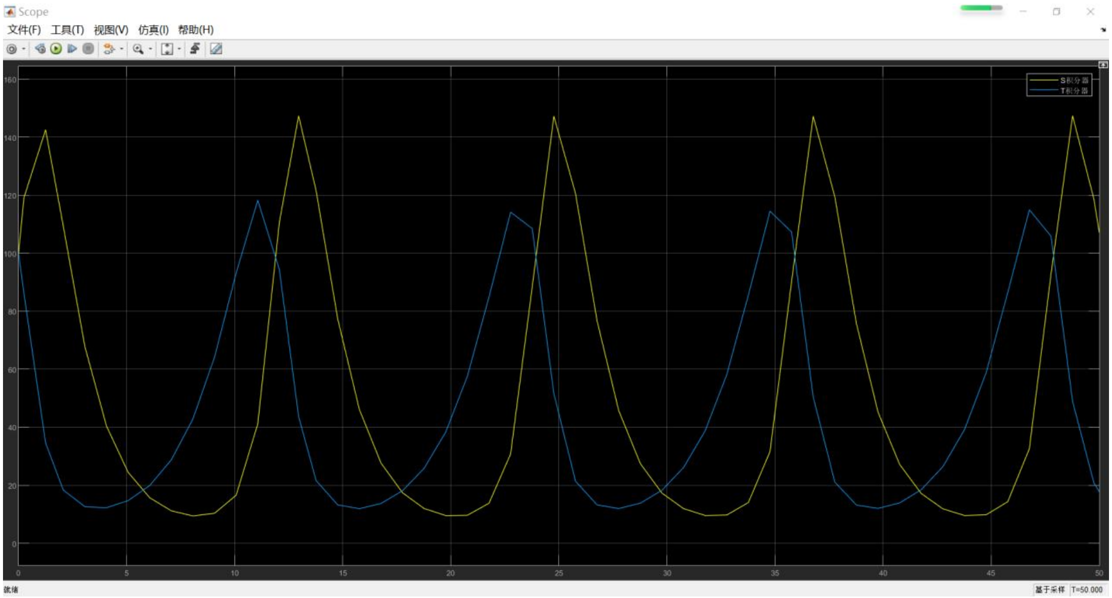


试用水印

微分方程初值在积分器中参数设置

仿真波形在 scope 中查看




Scope
文件(F) 工具(T) 视图(V) 仿真(I) 帮助(H)
S积分器
T积分器
160
140
120
100
80
60
40
20
0
0 5 10 15 20 25 30 35 40 45 50
就绪
基于采样
T=50.000

基本相符

试用水印