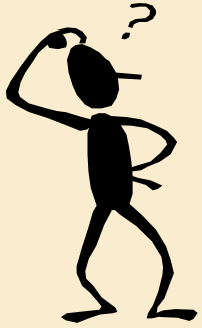
大学数学实验

Experiments in Mathematics

第3讲

数值计算II 常微分方程初值问题数值解

基本内容



1. 常微分方程实例，用微分方程建模

2. 两个最常用的数值算法:
   - 欧拉（Euler）方法
   - 龙格-库塔（Runge-Kutta）方法

3. 欧拉法和龙格-库塔方法的MATLAB实现

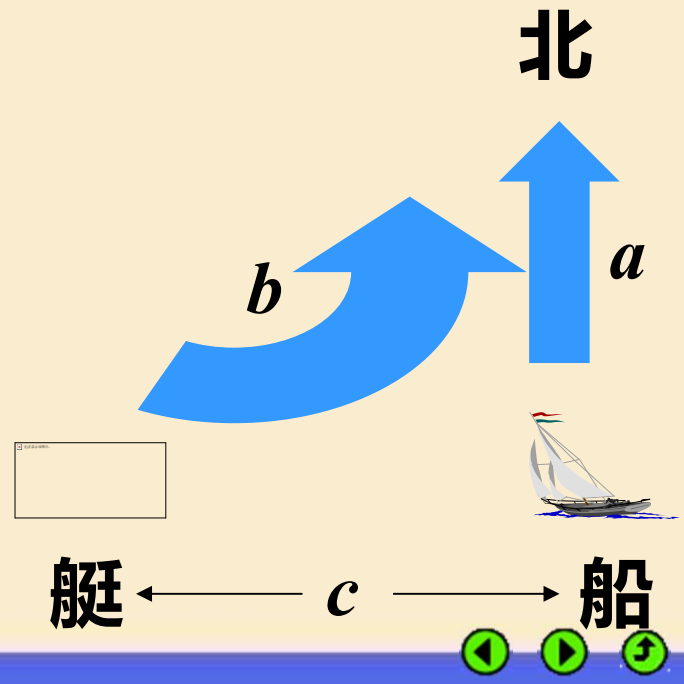
4. 实例求解

*5. 数值算法的收敛性、稳定性与刚性方程

实例1 海上缉私

海防某部缉私艇上的雷达发现正东方向*c*海里处有一艘走私船正以速度*a*向正北方向行驶，缉私艇立即以最大速度*b(>a)*前往拦截。如果用雷达进行跟踪时，可保持缉私艇的速度方向始终指向走私船。

- 建立任意时刻缉私艇位置及航线的数学模型,并求解;
- 求出缉私艇追上走私船的时间。



实例1 海上缉私

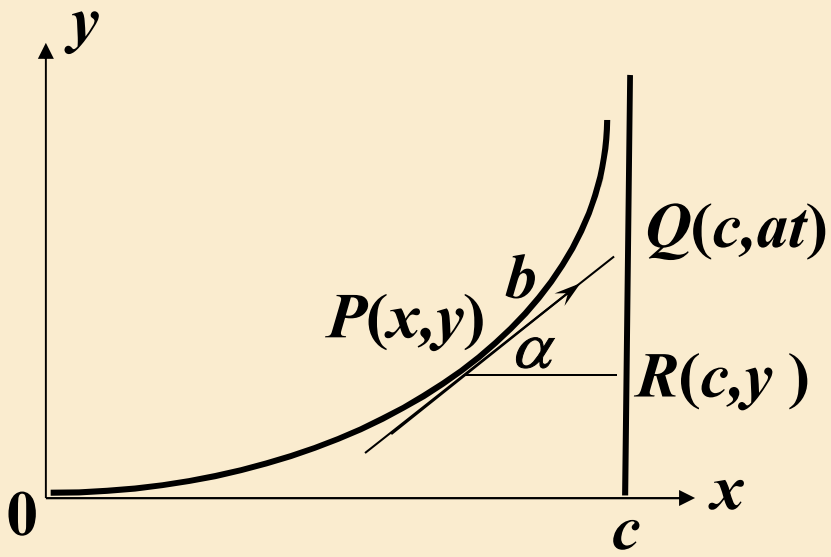
建立坐标系如图： t=0 艇在(0, 0), 船在(c, 0)； 船速a, 艇速b

时刻 t 艇位于$P(x, y)$, 船到达 $Q(c, at)$

模型：

$$\frac{dx}{dt}=b\cos\alpha,\frac{dy}{dt}=b\sin\alpha$$

$$\begin{cases}\dfrac{dx}{dt}=\dfrac{b(c-x)}{\sqrt{(c-x)^2+(at-y)^2}}\\[2ex]\dfrac{dy}{dt}=\dfrac{b(at-y)}{\sqrt{(c-x)^2+(at-y)^2}}\end{cases}$$



由方程$x(t), y(t)$的解析解不易推导

尝试借助数值解法求解

实例2　弱肉强食（捕食生态）

问题　自然界中同一环境下两个种群之间的生存方式

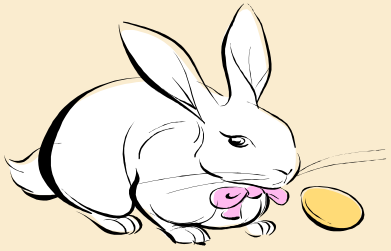
相互竞争　　相互依存　　弱肉强食

弱肉强食

种群甲靠丰富的自然资源生存　食饵(Prey)

种群乙靠捕食种群甲为生　捕食者(Predator)

两个种群的数量如何演变?



Lotka-Volterra 模型

Percentages of predators in the Fiume fish catch

1914	1915	1916	1917	1918	1919	1920	1921	1922	1923
12	21	22	21	36	27	16	16	15	11

In 1926 D'Ancona completed a statistical study of the numbers of each species sold on the fish markets of three ports: Fiume, Trieste, and Venice.

实例2　弱肉强食（捕食生态）

模型　食饵(甲) 的密度$x(t)$, 捕食者(乙)的密度$y(t)$

$\dot{x}/x = r,\ r>0$　甲独立生存的增长率r

$\dot{x}/x = r - ay,\ a>0$　乙使甲的增长率减小，减小量与 y 成正比

$\dot{y}/y = -d,\ d>0$　乙独立生存的死亡率d

$\dot{y}/y = -(d - bx),\ b>0$　甲使乙的死亡率减小，减小量与 x 成正比

$$\begin{cases} \dot{x} = (r-ay)x = rx - axy \\ \dot{y} = -(d-bx)y = -dy + bxy \\ x(0) = x_0, \quad y(0) = y_0 \end{cases}$$

Lotka-Volterra模型

$x(t), y(t)$无解析解

实例3　传染病传播的SIR模型

模型中把传染病流行范围内的人群分成S，I，R三类：

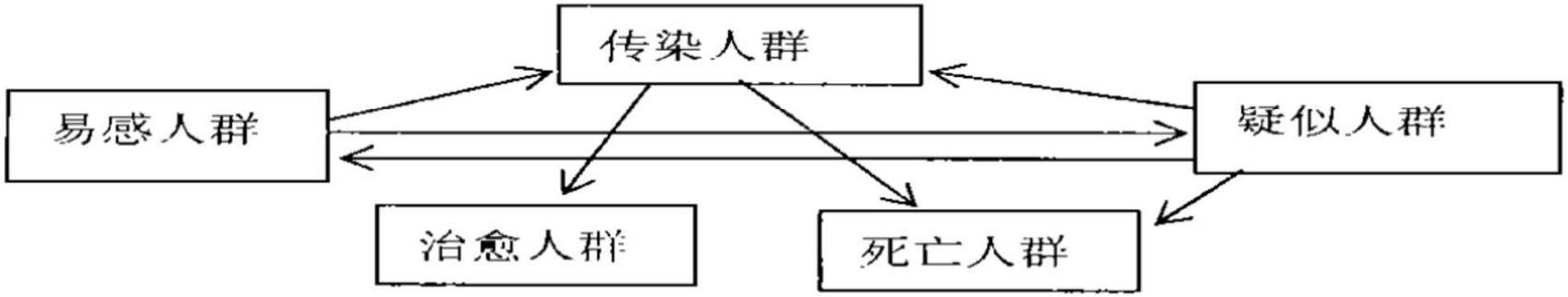
易感者（Susceptible），感病者（Infective），移出者（Removal）

1. 没有人进入或离开这个群体，同时不与群体外接触
2. 每个人或者是S（可能会染病），或者是I（已经染上并传播），或者是R（已感染过，并且不会再感染，包括死亡者）
3. 最初每个人或者是S，或者是I
4. 这种病平均持续α周, 则每周有β=1/α I移出，γ是S与I接触的发病率
5. 模型按照每周计算， S(t),I(t),R(t)

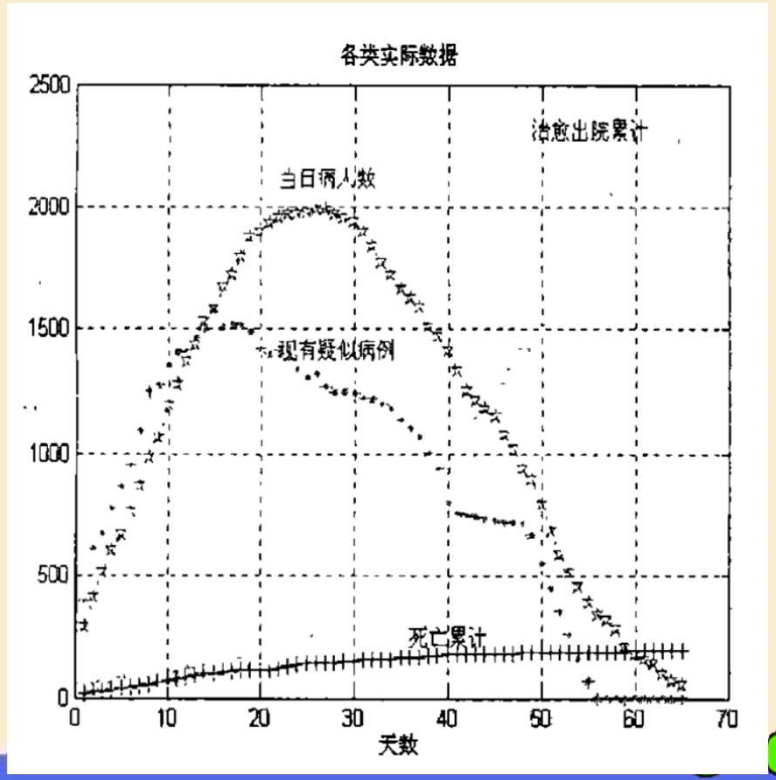
$$\begin{cases} \dfrac{dR}{dt} = \beta I(t) \\ \dfrac{dI}{dt} = -\beta I(t) + \gamma \cdot I(t) S(t) \\ \dfrac{dS}{dt} = -\gamma \cdot I(t) S(t) \end{cases} \qquad S(t) + I(t) + R(t) = \text{Constant} = N$$

$$I(0) = 5, S(0) = 995, R(0) = 0; \beta = 0.6, \gamma = 0.001407$$

SARS模型



$$
\begin{cases}
\dfrac{dY}{dt} = rH - (p+q)Y \\
\dfrac{dI}{dt} = sH + pY - dI - vI \\
\dfrac{dH}{dt} = qY - rH - sH \\
\dfrac{dD}{dt} = dI \\
\dfrac{dR}{dt} = vI \\
Y + H + I + D + R = N
\end{cases}
$$



“常微分方程初值问题数值解”的提法

$$\begin{cases} y' = f(x,y), \quad x \in [a,b], \\ y(x_0) = a, \end{cases}$$

当 $f(x,y)$ 满足一定连续性条件时，方程的解 $y(x)$ 存在唯一

不求解析解 $y = y(x)$　(无解析解或求解困难)

数值解：在一系列离散点 $a = x_0 < x_1 < x_2 < \cdots < x_n = b$

计算 $y(x_k)$ 的近似值 y_k

通常取等步长 h：　$x_n = x_0 + nh$

显式（向前） Euler 方法

$$y(x_{n+1}) = y(x_n + h) \approx y(x_n) + h\, y'(x_n) = y(x_n) + h\, f(x_n, y(x_n))$$

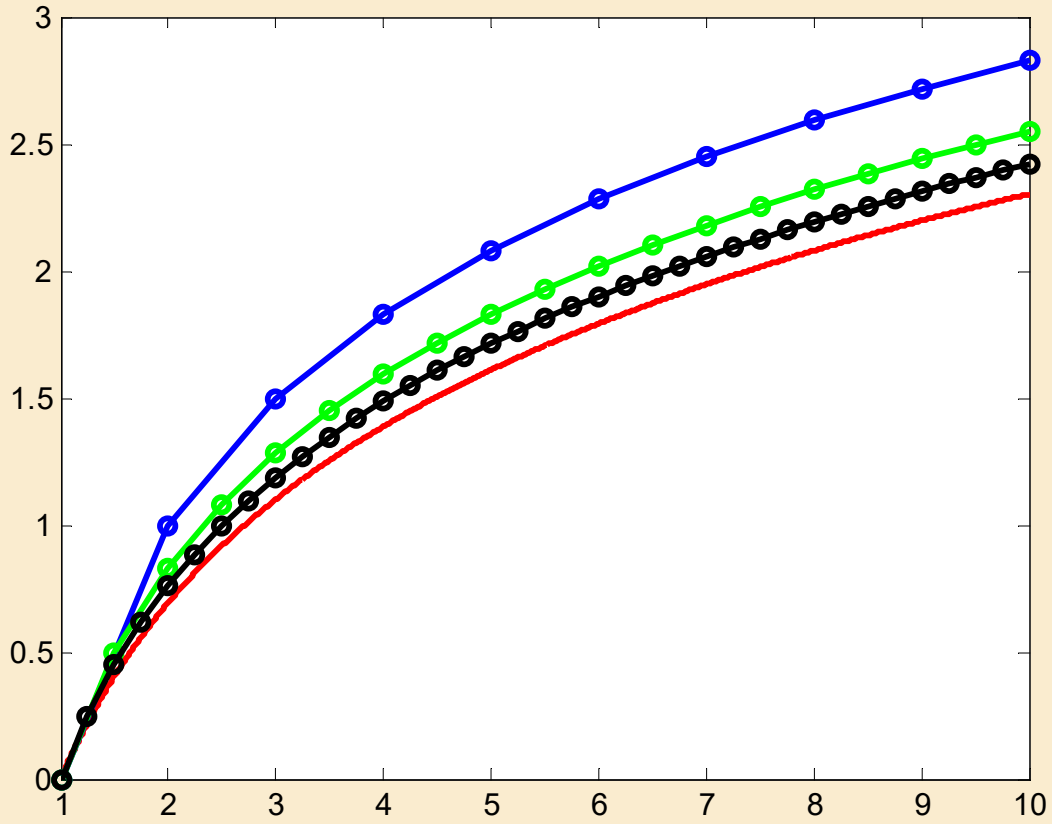
记 $y_0 = y(x_0) = a$, 取 $y_1 = y_0 + h\, f(x_0, y_0)$ 作为 $y(x_1)$ 的近似值
再以 $y_2 = y_1 + h\, f(x_1, y_1)$ 作为 $y(x_2)$ 的近似值

$$\begin{cases} y_{n+1} = y_n + h\, f(x_n, y_n), \quad n = 0,1,2,\cdots, \\ y_0 = a, \end{cases}$$

h：步长　　$x_n = x_0 + nh$

$y(x)$ 在 x_n 处的值记为 $y(x_n)$, 其近似值用 y_n 表示

向前欧拉法（欧拉折线法）



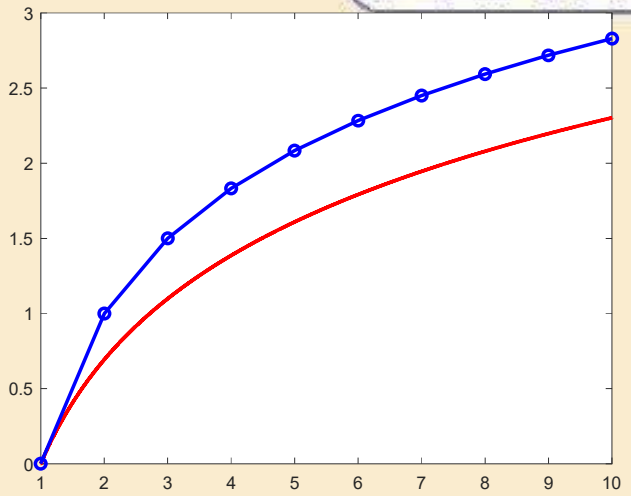
Exp03.m

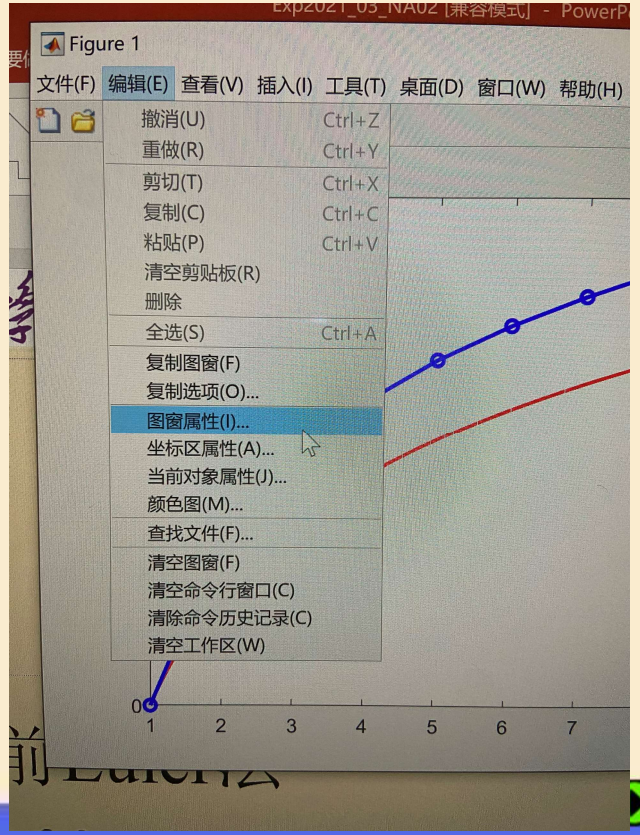
$$\begin{cases} y' = \dfrac{1}{x} & x \in [1,10] \\ y(1) = 0 \end{cases}$$

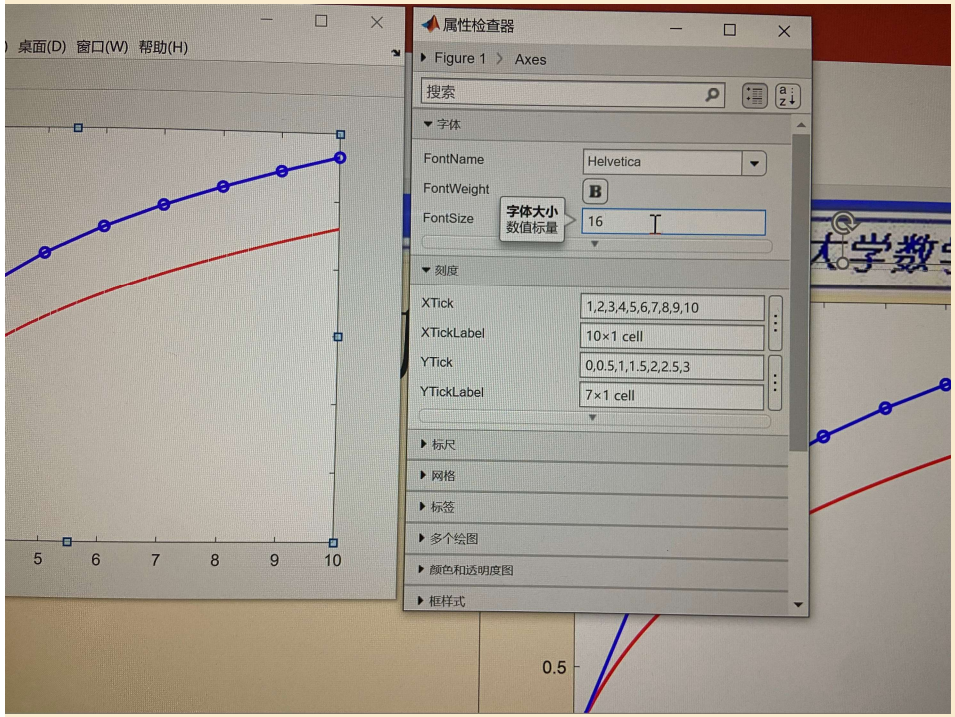
$$y(x) = \ln x$$

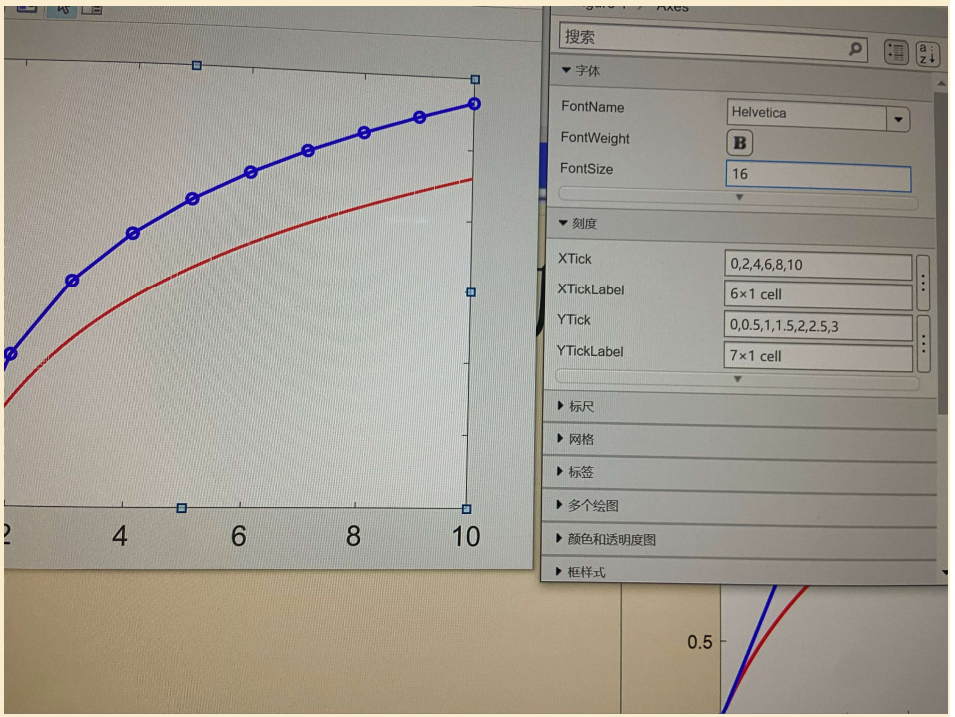
$$h = 1,\ 0.5,\ 0.25$$

```
% 向前Euler法
x=1:0.001:10;
plot(x,log(x),'r','LineWidth',2);
hold on;
h=1;
x=1:h:10;
y(1)=0;
for k=2:length(x)
    y(k)=y(k-1)+h/x(k-1);
end
plot(x,y,'bo-','LineWidth',2)
```






桌面(D) 窗口(W) 帮助(H)
属性检查器
Figure 1 Axes
搜索
字体
FontName
Helvetica
FontWeight
B
FontSize
字体大小
数值标量
16
刻度
XTick
1,2,3,4,5,6,7,8,9,10
XTickLabel
10×1 cell
YTick
0,0.5,1,1.5,2,2.5,3
YTickLabel
7×1 cell
标尺
网格
标签
多个绘图
颜色和透明度图
框样式
5
6
7
8
9
10
0.5


搜索
字体
FontName
Helvetica
FontWeight
B
FontSize
16
刻度
XTick
0,2,4,6,8,10
XTickLabel
6×1 cell
YTick
0,0.5,1,1.5,2,2.5,3
YTickLabel
7×1 cell
标尺
网格
标签
多个绘图
颜色和透明度图
框样式
4
6
8
10
0.5

隐式（向后）欧拉法

$$y(x_{n+1})-y(x_n)\approx y'(x_{n+1})(x_{n+1}-x_n)=f(x_{n+1},y(x_{n+1}))(x_{n+1}-x_n)$$

$$\begin{cases} y_{n+1}=y_n+h\,f(x_{n+1},y_{n+1}), & n=0,1,2,\cdots, \\ y_0=a, \end{cases}$$

隐式 Euler 方法, 或后退 Euler 方法

$$y_{n+1}^{(s+1)}=y_n+h\,f\left(x_{n+1},y_{n+1}^{(s)}\right), \quad s=0,1,2,\cdots$$

向前欧拉公式

$$y_{n+1} = y_n + hf(x_n, y_n)$$

向后欧拉公式

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

二者平均得到梯形公式

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})], n = 0,1,\cdots$$

仍为隐式公式，需迭代求解

改进欧拉公式　将梯形公式的迭代过程简化为两步

$$\bar{y}_{n+1} = y_n + hf(x_n, y_n)$$　预测

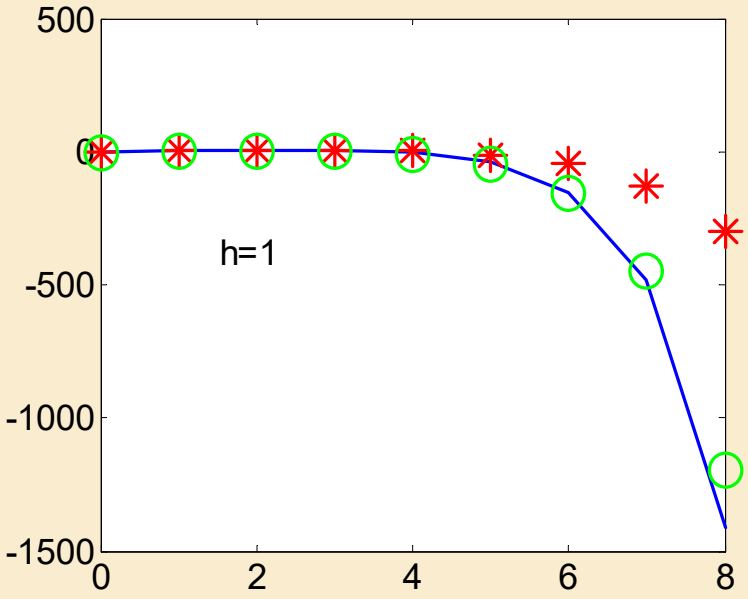
$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})]$$　校正

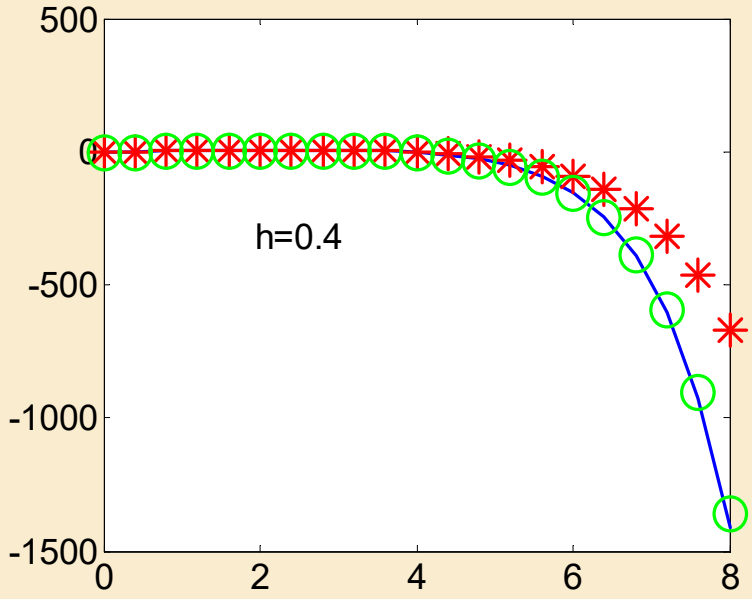
用显式 Euler 方法和改进 Euler 方法解初值问题

$$\begin{cases} y'(x) = y(x) - x^2 + 1 \\ y(0) = \dfrac{1}{2}, \end{cases}$$

Exp03.m

从 $x = 0$ 计算到 $x = 8$, 精确解 $y(x) = (x+1)^2 - \dfrac{1}{2}e^x$





```
clear; close;
h=1;  x=0:h:8;  y=(x+1).^2-exp(x)/2;    y1=1/2; y2=1/2;

n=length(x);
for k=2:n
    y1(k)=y1(k-1)+h*(y1(k-1)-x(k-1)^2+1);
end

for k=2:n
    yy=y2(k-1)+h*(y2(k-1)-x(k-1)^2+1);
    y2(k)=y2(k-1)+h*(y2(k-1)-x(k-1)^2+1+yy-x(k)^2+1)/2;
end

subplot(1,2,1);
plot(x,y,'LineWidth',2); hold on;
plot(x,y1,'R*','MarkerSize',16,'LineWidth',2);
plot(x,y2,'Go','MarkerSize',16,'LineWidth',2);
```

```
h=0.4; x=0:h:8; y=(x+1).^2-exp(x)/2; y1=1/2; y2=1/2;

n=length(x);
for k=2:n
    y1(k)=y1(k-1)+h*(y1(k-1)-x(k-1)^2+1);
end

for k=2:n
    yy=y2(k-1)+h*(y2(k-1)-x(k-1)^2+1);
    y2(k)=y2(k-1)+h*(y2(k-1)-x(k-1)^2+1+yy-x(k)^2+1)/2;
end

subplot(1,2,2);
plot(x,y,'LineWidth',2); hold on;
plot(x,y1,'R*','MarkerSize',16,'LineWidth',2);
plot(x,y2,'Go','MarkerSize',16,'LineWidth',2)
```

单 步 法

$$y_{n+1} = y_n + h\,\varphi(x_n, y_n, y_{n+1};\ h)$$

当 φ 中含有 y_{n+1} 时为隐式法

当 φ 中不含 y_{n+1} 时, 单步法为显式的,

$$y_{n+1} = y_n + h\,\varphi(x_n, y_n; h)$$

$\varphi(x, y; h)$ 称为增量函数

微分方程数值解法的误差分析

数值解法: 计算微分方程精确解$y(x_n)$的近似值y_n

按照步长h一步步计算, 每步都有误差;

每一步的误差会逐步积累， 称累积误差.

讨论计算一步出现的误差

假定$y_n = y(x_n)$, 估计y_{n+1}的误差: $y(x_{n+1}) - y_{n+1}$

局部截断误差

误差分析　估计欧拉公式的局部截断误差

$y(x_{n+1})$在x_n处作Taylor展开:

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + O(h^3)$$

向前欧拉公式 $y_{n+1} = y_n + hf(x_n, y_n)$

$y_n = y(x_n)$

$$y_{n+1} = y(x_n) + hf(x_n, y(x_n)) = y(x_n) + hy'(x_n)$$

$$T_{n+1} = y(x_{n+1}) - y_{n+1} = \frac{h^2}{2}y''(x_n) + O(h^3) = O(h^2)$$

局部截断误差主项为 $\frac{h^2}{2}y''(x_n)$

误差分析　估计欧拉公式的局部截断误差

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + O(h^3)$$

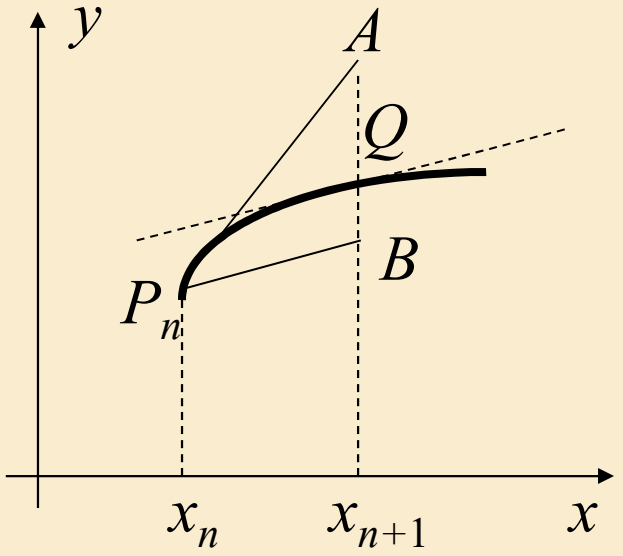
向后欧拉公式 $y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$

$$T_{n+1} = y(x_{n+1}) - y_{n+1} = -\frac{h^2}{2}y''(x_n) + O(h^3) = O(h^2)$$

局部截断误差主项为 $-\frac{h^2}{2}y''(x_n)$

梯形公式　向前、向后欧拉公式的平均

$$T_{n+1} = -\frac{h^3}{12}y'''(x_n) + O(h^4) = O(h^3)$$



精度

设 $y(x)$ 是初值问题 $\begin{cases} y' = f(x,y), & x \in (x_0, b], \\ y(x_0) = a, \end{cases}$ 的精确解,若 p 是满足 $y(x+h) - y(x) - h\,\varphi(x, y(x); h) = O(h^{p+1})$ 的最大正整数, 则称单步法 $y_{n+1} = y_n + h\,\varphi(x_n, y_n; h)$ 具有 p 阶精度, 或称该单步法是 p 阶方法.

设单步法是 p 阶方法, 将其局部截断误差写为

$$T_{n+1} = \psi(x_n, y(x_n))\,h^{p+1} + O(h^{p+2}),$$

则称 $\psi(x_n, y(x_n))\,h^{p+1}$ 为该单步法的 局部截断误差的主项, 或主局部截断误差 .

Euler和隐式Euler是1阶方法，梯形和改进Euler法是2阶方法

龙格-库塔方法

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))\,dx = y(x_n) + h\, f(\xi, y(\xi))$$

- 向前，向后欧拉公式：
  用$[x_n, x_{n+1}]$内 1 个点的导数估计前进方向
- 梯形公式，改进欧拉公式：
  用$[x_n, x_{n+1}]$内 2 个点导数的平均值估计前进方向

龙格-库塔方法的基本思想

在$[x_n, x_{n+1}]$内多取几个点，将它们的导数加权平均估计前进方向，设法构造出精度更高的计算公式。

Runge-Kutta 方法

用 **Taylor** 展开构造高阶数值方法

1. 确定 c_1,使 $y_{n+1}=y_n+hc_1\,f(x_n,y_n)$ 具有尽可能高阶的精度

$\Rightarrow c_1=1$, 此时局部截断误差主项为 $\dfrac{h^2}{2}y''(x_n)$

$y_{n+1}=y_n+h\,f(x_n,y_n)$ 即为显式 **Euler** 方法

2. 确定 $c_1\ c_2\ a_2\ b_{21}$,使 $y_{n+1}=y_n+hc_1K_1+hc_2K_2$ 具有尽可能高阶的精度

其中 $K_1=f(x_n,y_n)$

$K_2=f(x_n+a_2h\ ,y_n+hb_{21}K_1)$

二级二阶Runge-Kutta方法

$$\begin{cases} y_{n+1} = y_n + hc_1K_1 + hc_2K_2 \\ K_1 = f(x_n, y_n) \qquad (n = 0,1,2,\ldots) \\ K_2 = f(x_n + a_2h, y_n + hb_{21}K_1) \end{cases}$$

适当选择参数 c_1, c_2, a_2, b_{21}, 使局部截断误差

$T_{n+1} = y(x_{n+1}) - y_{n+1} = O(h^3)$,这里仍假定 $y_n = y(x_n)$。

由二元函数Taylor展开式:

$$K_2 = f(x_n, y_n) + a_2hf'_x(x_n, y_n) + hb_{21}K_1f'_y(x_n, y_n) + O(h^2)$$

$$= y'(x_n) + h(a_2f'_x(x_n, y_n) + b_{21}f(x_n, y_n)f'_y(x_n, y_n)) + O(h^2)$$

于是 $y_{n+1} = y(x_n) + (c_1 + c_2)hy'(x_n) + [a_2c_2f'_x(x_n, y_n)$

$$+ b_{21}c_2f(x_n, y_n)f'_y(x_n, y_n)]h^2 + O(h^3)$$

$$y''(x) = \frac{dy(x)}{dx} = \frac{df(x, y(x))}{dx} = f'_x(x, y(x)) + \frac{dy(x)}{dx} \cdot f'_y(x, y(x))$$

$$\begin{cases} c_1 + c_2 = 1 \\ c_2a_2 = \frac{1}{2} \\ c_2b_{21} = \frac{1}{2} \end{cases}$$

2级 Runge-Kutta 方法(2阶)

$$\begin{cases} c_1 + c_2 = 1 \\ c_2 a_2 = \dfrac{1}{2} \\ c_2 b_{21} = \dfrac{1}{2} \end{cases}$$ 时, 上述公式具有 $O(h^3)$ 的局部截断误差

$$\begin{cases} y_{n+1} = y_n + hc_1K_1 + hc_2K_2 \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + a_2 h, y_n + hb_{21}K_1) \end{cases}$$

取 $a_2 = \dfrac{1}{2}$, 得 $c_2 = 1, c_1 = 0, b_{21} = \dfrac{1}{2}$

$$\Rightarrow y_{n+1} = y_n + hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(x_n, y_n)\right)$$ 中点公式

取 $a_2 = 1$, 得 $c_2 = \dfrac{1}{2}, c_1 = \dfrac{1}{2}, b_{21} = 1$

$$\Rightarrow y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n))\right]$$ 改进Euler方法

龙格-库塔方法的一般形式

$$
\begin{cases}
y_{n+1} = y_n + h\sum_{i=1}^{L} \lambda_i K_i \\
K_1 = f(x_n, y_n) \\
K_2 = f(x_n + c_2 h, y_n + c_2 h K_1) \\
\cdots \\
K_i = f(x_n + c_i h, y_n + c_i h \sum_{j=1}^{i-1} a_{ij} K_j), \qquad i = 3, 4, \cdots, L
\end{cases}
$$

L级?阶　　使精度尽量高

λ_i, c_i, a_{ij}满足 $\sum_{i=1}^{L} \lambda_i = 1, 0 \le c_i \le 1, \sum_{j=1}^{i-1} a_{ij} = 1$

经典 Runge-Kutta 方法（4级4阶）

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{1}{6}h\left(K_1 + 2K_2 + 2K_3 + K_4\right) \\
K_1 = f\left(x_n, y_n\right) \\
K_2 = f\left(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}hK_1\right) \\
K_3 = f\left(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}hK_2\right) \\
K_4 = f\left(x_n + h, y_n + hK_3\right)
\end{cases}
$$

误差分析

算法精度的阶的定义

一个算法的局部截断误差为$O(h^{p+1})$

⇩

该算法具有p阶精度

	局部截断误差	精度
向前欧拉公式	$O(h^2)$	1阶
向后欧拉公式	$O(h^2)$	1阶
梯形公式	$O(h^3)$	2阶
改进欧拉公式	$O(h^3)$	2阶
经典龙格-库塔公式	$O(h^5)$	4阶

微分方程组和高阶方程初值问题的数值解

欧拉方法和龙格-库塔方法可直接推广到微分方程组

$$\begin{cases} y' = f(x,y,z) \\ z' = g(x,y,z) \\ y(x_0) = y_0,\ z(x_0) = z_0 \end{cases}$$

向前欧拉公式

$$\begin{cases} y_{n+1} = y_n + hf(x_n,y_n,z_n) \\ z_{n+1} = z_n + hg(x_n,y_n,z_n) \\ n = 0,1,2,\cdots \end{cases}$$

高阶方程需要先降阶为一阶微分方程组

$$y'' = f(x,y,y') \quad \xrightarrow{y_1 = y} \quad \begin{cases} y_1' = y_2 \\ y_2' = f(x,y_1,y_2) \end{cases}$$

龙格—库塔方法的 MATLAB 实现

$$\dot{x}(t) = f(t,x),\ x(t_0) = x_0,\ x = (x_1,\cdots x_n)^T,\ f = (f_1,\cdots f_n)^T$$

[t,x]=ode23(@f, ts, x0)　3阶龙格-库塔公式

[t,x]=ode45(@f, ts, x0)　5阶龙格-库塔公式

f是待解方程写成的函数m文件：

```
function dx=f(t,x)
dx=[f1; f2;…; fn];
```

ts = [t0,t1, …,tf]　输出指定时刻 **t0,t1, …,tf** 的函数值

ts = t0:k:tf　输出 **[t0,tf]** 内等分点处的函数值

x0为函数初值(**n**维)　输出**t=ts, x**为相应函数值(**n**维)

缺省精度(相对误差10^{-3}，绝对误差10^{-6})，计算步长按精度要求自动调整.

Matlab 函数 ode23

A 3(2) pair of Runge-Kutta formulas, *Appl. Math. Letters*, Vol.2,1989, 321-325

1989 paper by Bogacki and Shampine and a 1994 textbook by Shampine. The "23" in the function names indicates that two simultaneous single step formulas, one of second order and one of third order, are involved.

The method has three stages, but there are four slopes s_i because, after the first step, the s_1 for one step is the s_4 from the previous step. The essentials are

$$
\begin{aligned}
s_1 &= f(t_n, y_n) \\
s_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} s_1) \\
s_3 &= f(t_n + \frac{3}{4}h, y_n + \frac{3}{4}h s_2) \\
t_{n+1} &= t_n + h \\
y_{n+1} &= y_n + \frac{h}{9}(2s_1 + 3s_2 + 4s_3) \\
s_4 &= f(t_{n+1}, y_{n+1})
\end{aligned}
$$

$$y_{n+1} = y_n + h\sum_{i=1}^{4} b_i s_i \qquad \frac{7}{24} \quad \frac{1}{4} \quad \frac{1}{3} \quad \frac{1}{8}$$

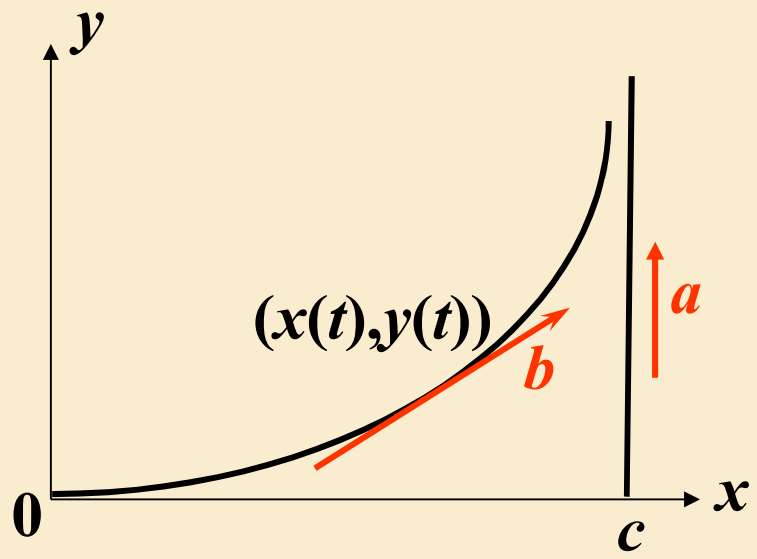
实例1　海上缉私(续)

模型的数值解

$$\begin{cases} \dfrac{dx}{dt} = \dfrac{b(c-x)}{\sqrt{(c-x)^2+(at-y)^2}} \\ \dfrac{dy}{dt} = \dfrac{b(at-y)}{\sqrt{(c-x)^2+(at-y)^2}} \end{cases}$$

$$x(0)=0, \quad y(0)=0$$

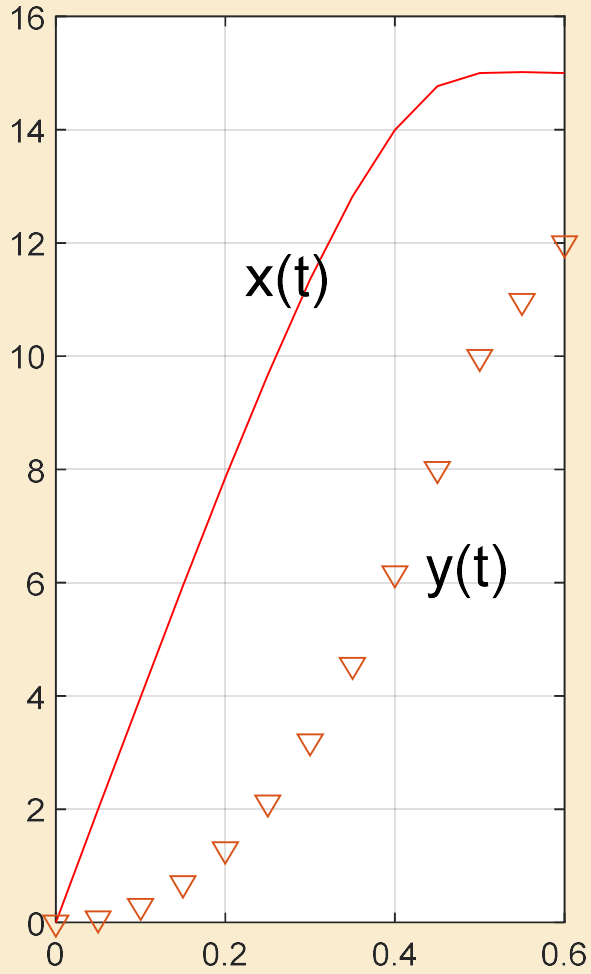
设: 船速a=20 (海里/小时)
艇速b=40 (海里/小时)
距离c=15 (海里)

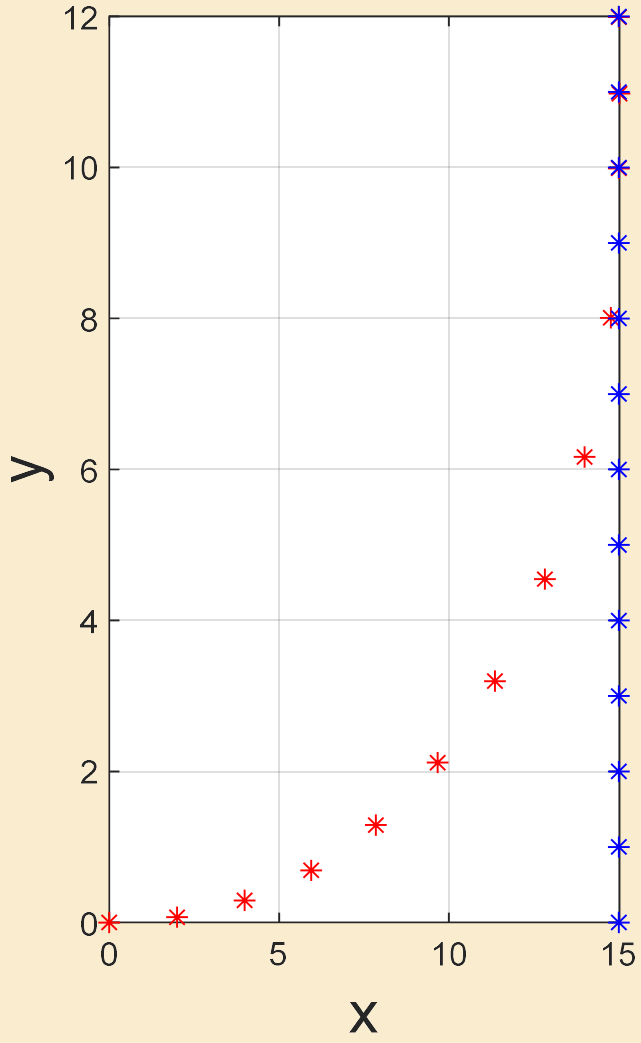


求: 缉私艇的位置$x(t)$,$y(t)$
缉私艇的航线$y(x)$

```
clear,close;
%Set the definied time
ts=0:0.05:.6;

x0=[0 0];a=20;b=40;c=15;
opt=odeset('RelTol',1e-6,'AbsTol',1e-9);

[t,x]=ode23(@jisi,ts,x0,[],a,b,c);  y1=a*t;

% draw x(t),y(t)
subplot(1,2,1);  plot(t,x(:,1),'r'),
grid on,hold on;   gtext('x(t)','FontSize',16),
plot(t,x(:,2),'v'); text('y(t)','FontSize',16),pause

%draw y(x): the position of tatch jisi
subplot(1,2,2); plot(x(:,1),x(:,2),'r*'),
grid on, hold on,
xlabel('x','FontSize',16),ylabel('y','FontSize',16)
plot(15*ones(length(x),1),y1,'b*')
```

Exp03.m
jisi.m





```
% 显示追击过程
close;
plot(x(1,1),x(1,2),'r*'); hold on;
plot(15,y1(1),'bo');
axis([0 16 0 12]);  pause
for k=2:length(x)
    plot(x(k,1),x(k,2),'r*');
    plot(15,y1(k),'bo');
    plot(x(k-1,1),x(k-1,2),'g*');
    plot(15,y1(k-1),'yo');pause;
end
```

实例1　海上缉私(续)

模型的数值解

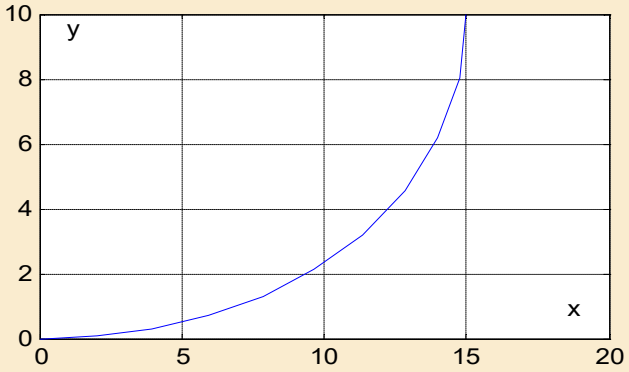
a=20, b=40, c=15

走私船的位置

$x_1(t)= c$=15

$y_1(t)=at$=20t

缉私艇的航线$y(x)$



t=0.5时缉私艇追上走私船

t	$x(t)$	$y(t)$	$y_1(t)$
0	0	0	0
0.05	1.9984	0.0698	1.0
0.10	3.9854	0.2924	2.0
0.15	5.9445	0.6906	3.0
0.20	7.8515	1.2899	4.0
0.25	9.6705	2.1178	5.0
0.30	11.3496	3.2005	6.0
0.35	12.8170	4.5552	7.0
0.40	13.9806	6.1773	8.0
0.45	14.7451	8.0273	9.0
0.50	15.0046	9.9979	10.0

实例1

海上缉私(续)

设b，c不变，a变大为30, 35, …接近40, 观察解的变化：

a=35, b=40, c=15

t=? 缉私艇追上走私船

累积误差较大

提高精度!

t	$x(t)$	$y(t)$	$y_1(t)$
0	0	0	0
0.1	3.9561	0.5058	3.5
0.2	7.5928	2.1308	7.0
0.3	10.5240	4.8283	10.5
0.4	12.5384	8.2755	14.0
0.5	13.7551	12.0830	17.5
…	…	…	…
1.2	14.9986	40.0164	42.0
1.3	14.9996	44.0165	45.5
1.4	15.0117	48.0183	49.0
1.5	15.0023	52.0146	52.5
1.6	14.9866	55.9486	56.0

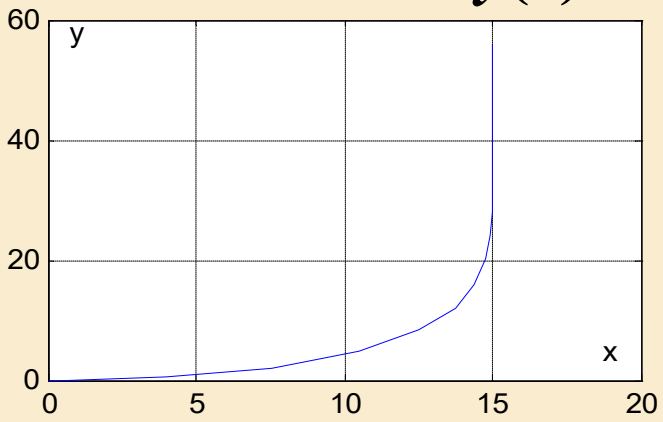
实例1

海上缉私（续）

```
opt=odeset('RelTol',1e-6,
                'AbsTol',1e-9);
[t,x]=ode45(@jisi,ts,x0,opt);
```

a=35, b=40, c=15

缉私艇的航线$y(x)$



t=1.6时缉私艇追上走私船

判断“追上”的有效方法?

模型的数值解

t	$x(t)$	$y(t)$	$y_1(t)$
0	0	0	0
0.1	3.956104	0.505813	3.5
0.2	7.592822	2.130678	7.0
0.3	10.521921	4.829308	10.5
0.4	12.539454	8.269840	14.0
0.5	13.753974	12.075344	17.5
…	…	…	…
1.2	14.999616	40.000005	42.0
1.3	14.999963	44.000005	45.5
1.4	14.999993	48.000005	49.0
1.5	14.999998	52.000005	52.5
1.6	15.000020	55.999931	56.0

实 例 2　　弱 肉 强 食

模型　食饵(甲) 的密度$x(t)$, 捕食者(乙)的密度$y(t)$

$$\dot{x}/x = r,\ r>0$$

甲独立生存的增长率r

$$\dot{x}/x = r - ay,\ a>0$$

乙使甲的增长率减小,
减小量与 y 成正比

$$\dot{y}/y = -d,\ d>0$$

乙独立生存的死亡率d

$$\dot{y}/y = -(d - bx),\ b>0$$

甲使乙的死亡率减小,
减小量与 x 成正比

$$\begin{cases} \dot{x} = (r-ay)x = rx - axy \\ \dot{y} = -(d-bx)y = -dy + bxy \\ x(0) = x_0, \quad y(0) = y_0 \end{cases}$$

Lotka-Volterra模型

$x(t)$, $y(t)$无解析解

实例2 弱肉强食 模型的数值解

$$\begin{cases} \dot{x} = rx - axy \\ \dot{y} = -dy + bxy \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

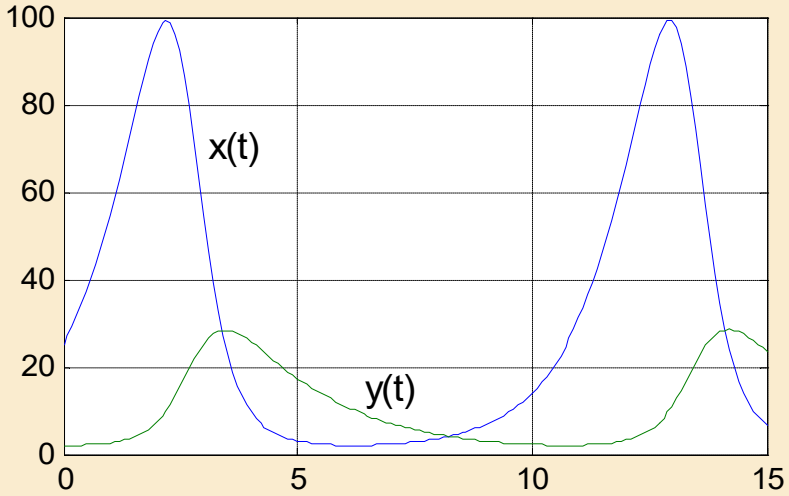
$r = 1, d = 0.5$

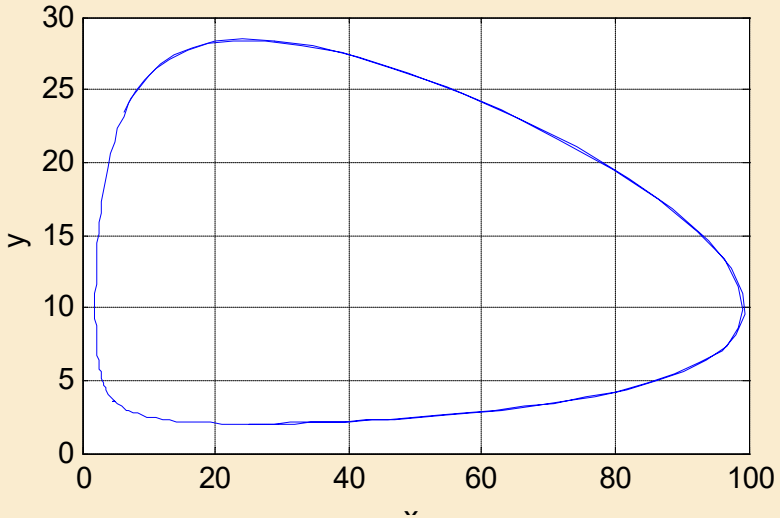
$a = 0.1, b = 0.02$

$x_0 = 25, y_0 = 2$

r:x自然出生率
d:y自然死亡率

Exp03, Shier.m

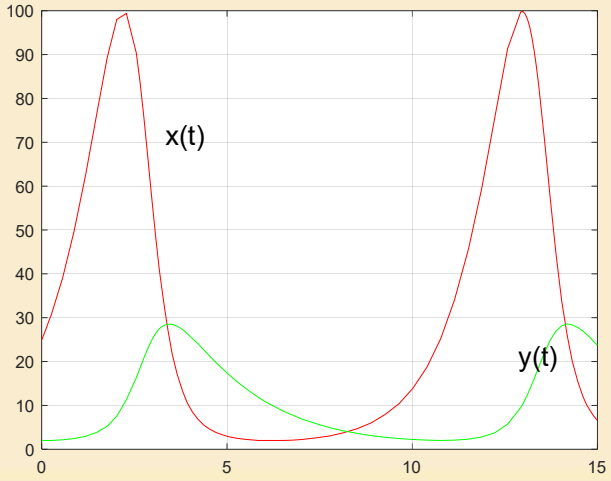


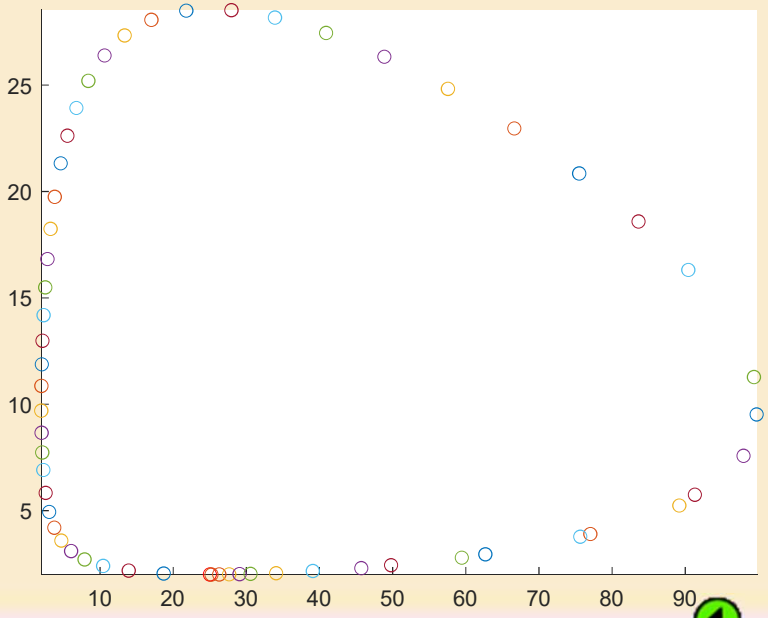


猜测 $x(t), y(t)$是周期函数; $y(x)$是封闭曲线

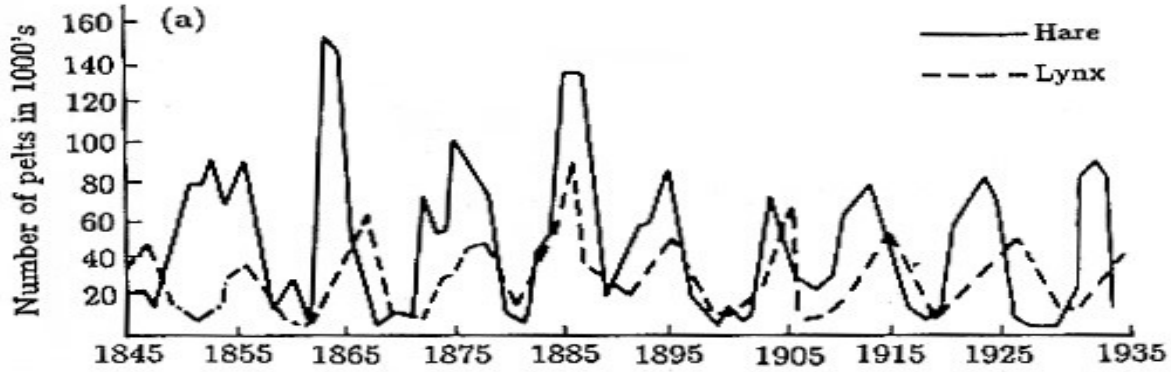
数值积分计算一个周期的平均值: $\bar{x} \approx 25,\ \bar{y} \approx 10$

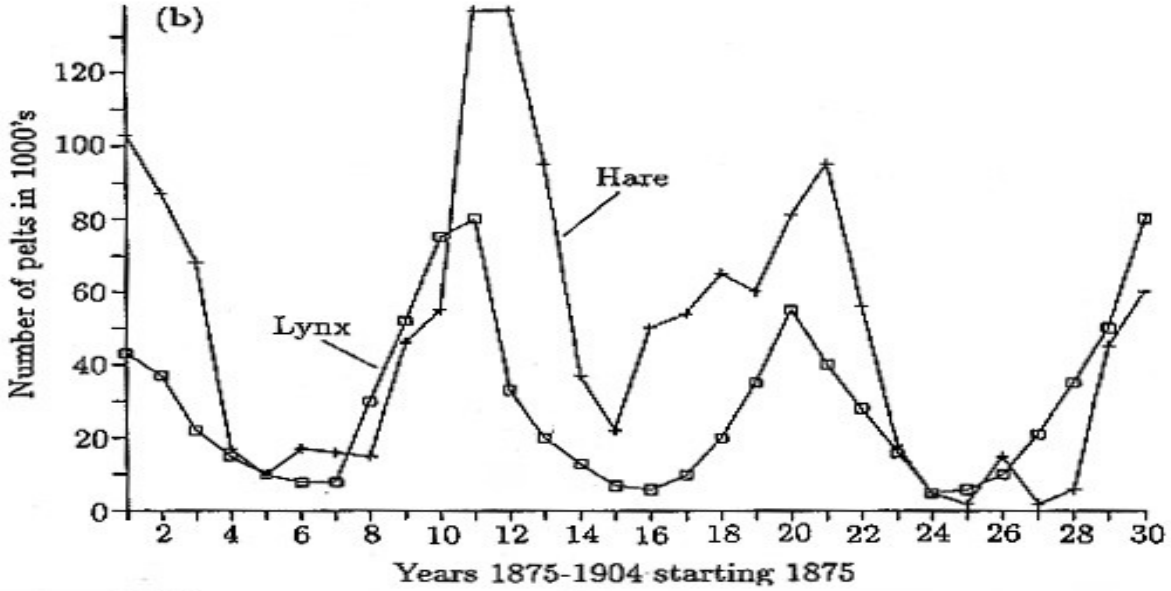
```
clear; close;
[t,x]=ode45(@Shier,[0 15],[25 2]);
plot(t,x(:,1),'r'),grid,hold on;
gtext('x(t)','FontSize',16),
plot(t,x(:,2),'g'),  gtext('y(t)','FontSize',16);
pause; close
axis([min(x(:,1)) max(x(:,1)) min(x(:,2))
max(x(:,2))]); hold on; plot(x(1,1),x(1,2),'ro');
for k=2:length(x)
    plot(x(k,1),x(k,2),'o');
    pause;
end
```

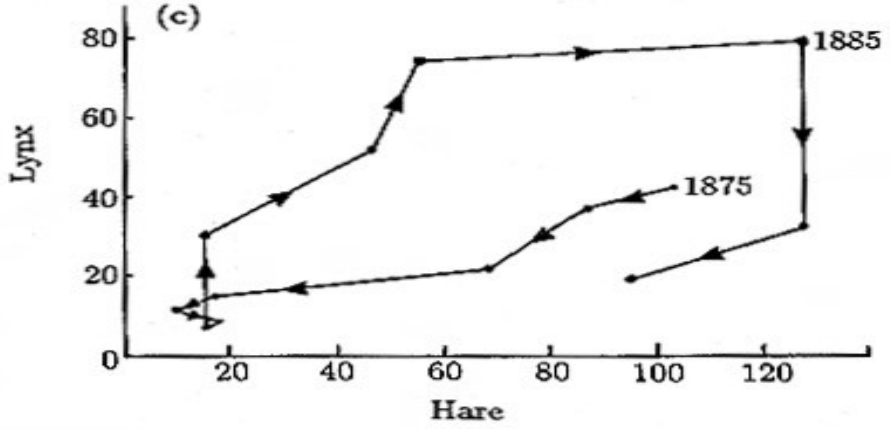




加拿大
山猫
野兔


(a)
Hare
Lynx
Number of pelts in 1000's
160
140
120
100
80
60
40
20
1845
1855
1865
1875
1885
1895
1905
1915
1925
1935


(b)
Hare
Lynx
Number of pelts in 1000's
120
100
80
60
40
20
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
Years 1875-1904 starting 1875


(c)
1885
1875
Lynx
80
60
40
20
0
20
40
60
80
100
120
Hare

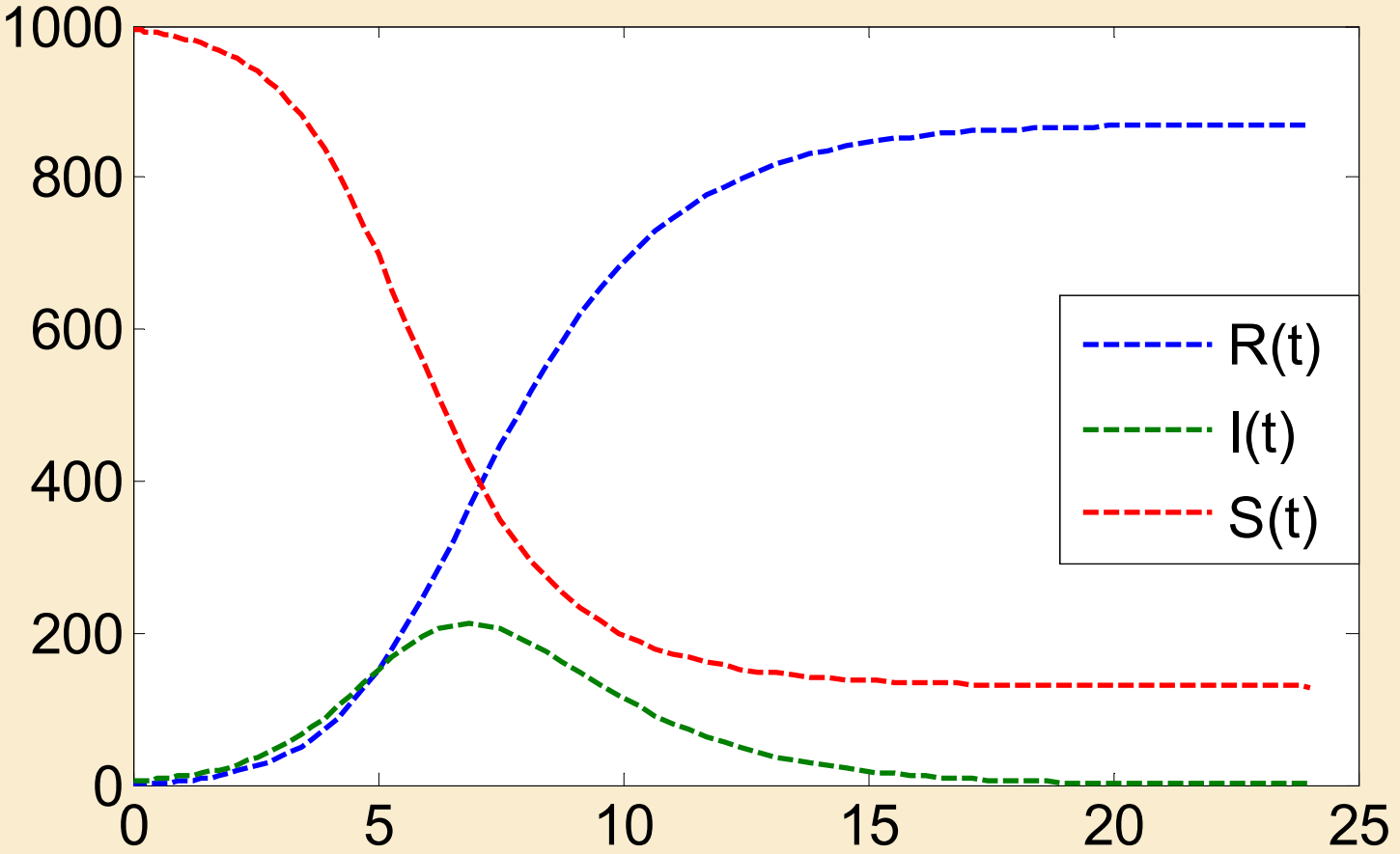
实例3　传染病传播的SIR模型

$$\begin{cases} \dfrac{dR}{dt} = \beta I(t) \\ \dfrac{dI}{dt} = -\beta I(t) + \gamma \cdot I(t) S(t) \\ \dfrac{dS}{dt} = -\gamma \cdot I(t) S(t) \end{cases}$$

$$S(t) + I(t) + R(t) = \text{Constant} = N$$

$$I(0) = 5, S(0) = 995, R(0) = 0; \beta = 0.6, \gamma = 0.001407$$

```
function dy = fsir(t,y)
dy = zeros(3,1);     % a column vector
dy(1) = 0.6*y(2);
dy(2) = -0.6*y(2)+0.001407*y(2)*y(3);
dy(3) = -0.001407*y(2)*y(3);

options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);
[T,Y] = ode45(@fsir,[0 24],[0 5 995],options);
plot(T,Y)
```



易感者（**Susceptible**），

感病者（**Infective**），

移出者（**Removal**）

布置实验

目的

1.用MATLAB软件掌握求微分方程数值解的方法，并对结果作初步分析；

2.通过实例学习用微分方程模型解决简化的实际问题。

内容

见网络学堂--课程作业