常微分方程及数值解

本次课主要内容

- ODE的精确解—dsolve('eqn1', 'eqn2', ..., t)
  - 通解
  - 特解
- ODE的数值解—[t, y]=ode45('odefun', tspan, y0, options)
  - Euler法
  - Euler法的改进
  - 龙格-库塔法

常微分方程

常微分方程是一种包含自变量、函数、函数的导数和高阶导数的等式. 常微分方程（组）经常被用于描述事物的动态变化，如经济波动、物体运动轨迹、受力形变等.

下面介绍如何用Matlab来求解常微分方程.

求常微分方程（组）的通解 dsolve('eqn1', 'eqn2',..., 't')

eqn1、eqn2 等为各个方程，t为变量.

例1. 求解$x'(t)+4x(t)=\sin 2t$.

```
dsolve('Dx+4*x=sin(2*t)')

ans =

-1/10*cos(2*t)+1/5*sin(2*t)+exp(-4*t)*C1
```

练习：求解$x'(t)+4x(t)=\cos 2t$; $\dfrac{dy}{dx}+2y(x)=\sin 2x$.

常微分方程

注意，系统默认自变量为t，其它自变量必须注明！

比如对于方程：$x'(t)=t$.　　　　$y'(x)=x$.

其解分别应为：$x(t)=\dfrac{t^2}{2}+C$,　　$y(x)=\dfrac{x^2}{2}+C$,

输入命令：>> dsolve('Dx=t')　　>> dsolve('Dy=x')

得到的是：　ans = 1/2*t^2+C1　　ans = x*t+C1

对于后者，正确的求解命令应为：>> syms x;

```
dsolve('Dy=x', 'x')
```

常微分方程

例2. 求解$x''(t)+2x'(t)-3x(t)=e^t$.

```
dsolve('D2x+2*Dx-3*x=exp(t)')
```

```
ans = 1/4*t*exp(t)-1/16*exp(t)+C1*exp(t)+C2*exp(-3*t)
```

练习：$x''(t)+2x'(t)-\cos x(t)=\sin x(t)e^t$

```
dsolve('D2x+2*Dx-cosx=sin(x)*exp(t)')
```

```
Warning: Explicit solution could not be found.
> In H:\Matlab6p5\toolbox\symbolic\dsolve.m at line 326

ans =

[ empty sym ]
```

例3. 求解常微分方程组：$\begin{cases} x'(t)=-3y(t) \\ y'(t)=4x(t) \end{cases}$

输出的结果是向量函数，代码为：

```
Syms t;
[x, y]=dsolve('Dx=-3*y', 'Dy=4*x', 't')
```

```
x = cos(2*3^(1/2)*t)*C1-1/2*3^(1/2)*sin(2*3^(1/2)*t)*C2

y = 2/3*3^(1/2)*sin(2*3^(1/2)*t)*C1+cos(2*3^(1/2)*t)*C2
```

练习：$\begin{cases} x'=2x-3y \\ y'=4x-5y \end{cases}$

求常微分方程（组）的特解

常微分方程的通解均带有任意常数项，要确定常数项的取值就需要定解条件.

定解条件包括初值条件、边值条件，在此基础上求解常微分方程的解，可以用命令：

```
dsolve('eqn', 'condition1', ..., 'conditionk', t)
```

其中eqn为方程，condition1等为定解条件.

例4. 求解常微分方程初值问题：$\begin{cases} y''(x)+3y'(x)+2y(x)=\sin 2x, \\ y(0)=1,\ y'(0)=1. \end{cases}$

```
dsolve('D2y+3*Dy+2*y=sin(2*x)','y(0)=1','Dy(0)=1','x')

ans =

-3/20*cos(2*x)-1/20*sin(2*x)+17/5*exp(-x)-9/4*exp(-2*x)
```

练习：$y''(x)+2y'(x)-3y(x)=e^x,\ y(0)=0,\ y(1)=0$

常微分方程

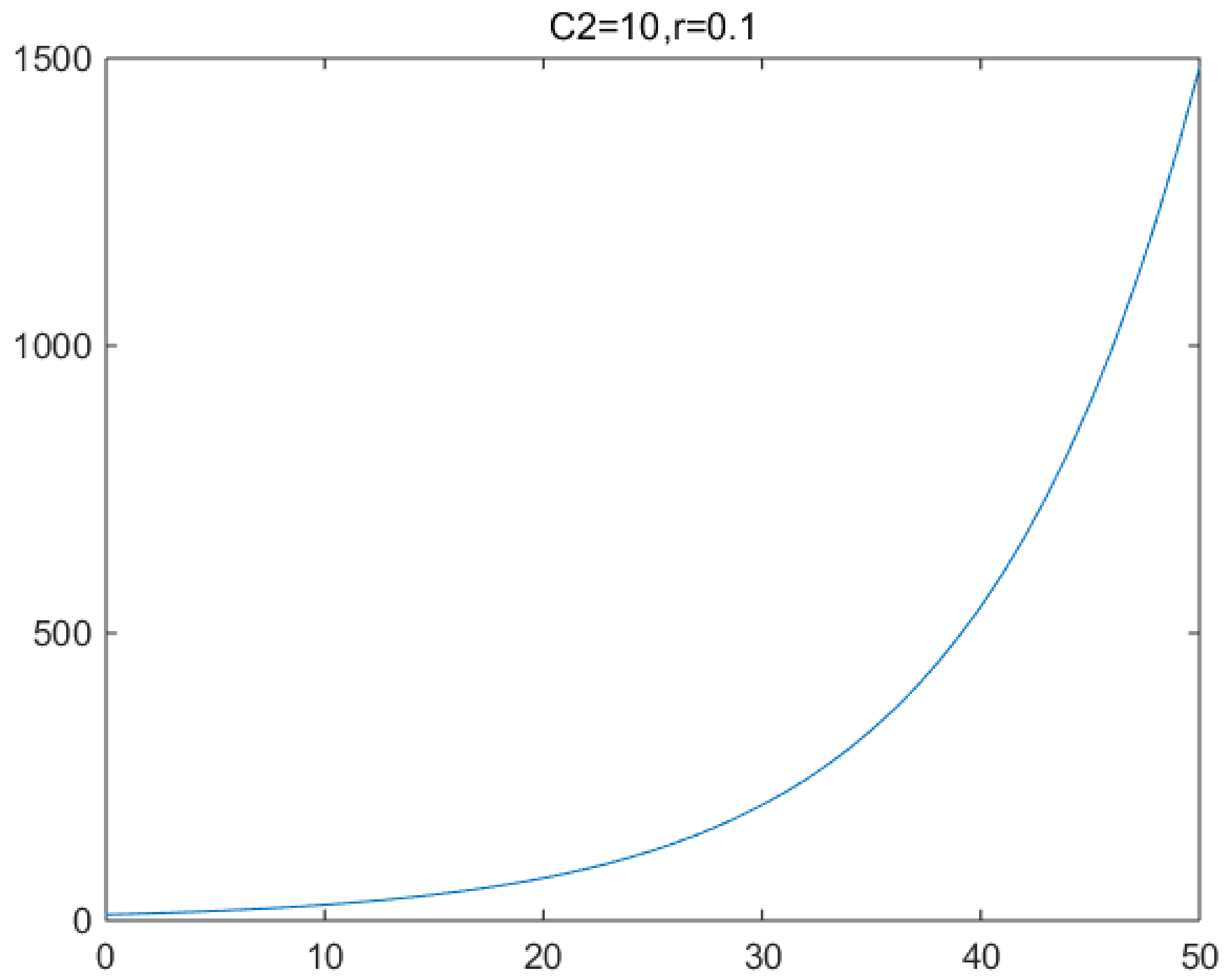
例5. 种群规模研究经常用到常微分方程. 比如最简单的指数增长模型：设 t 时刻种群总数为 $x(t)$，种群增速与现有生物总数成正比， 比例系数为 r (相对增长率).

据此可以建立常微分方程模型： $x'(t)=r\cdot x(t)$.

求其通解可得： $x=C\cdot e^{rt}$.

```
s=dsolve('Dx=r*x','t')
```

```
s = C1*exp(r*t)
```



常微分方程

考虑到环境和资源的限制，种群规模又会对种群增长产生一定的阻滞作用．假设种群的相对增长率是总数的减函数：$r = a(1 - \frac{x}{N})$，N是环境可容纳的种群数量上限．

据此建立阻滞增长模型：　$x'(t) = a(1 - \frac{x}{N}) \cdot x.$

此方程有三类解：

```
s=dsolve('Dx=a*(1-x/N)*x','t')
```

两个常值解：$x(t) \equiv N$，0；

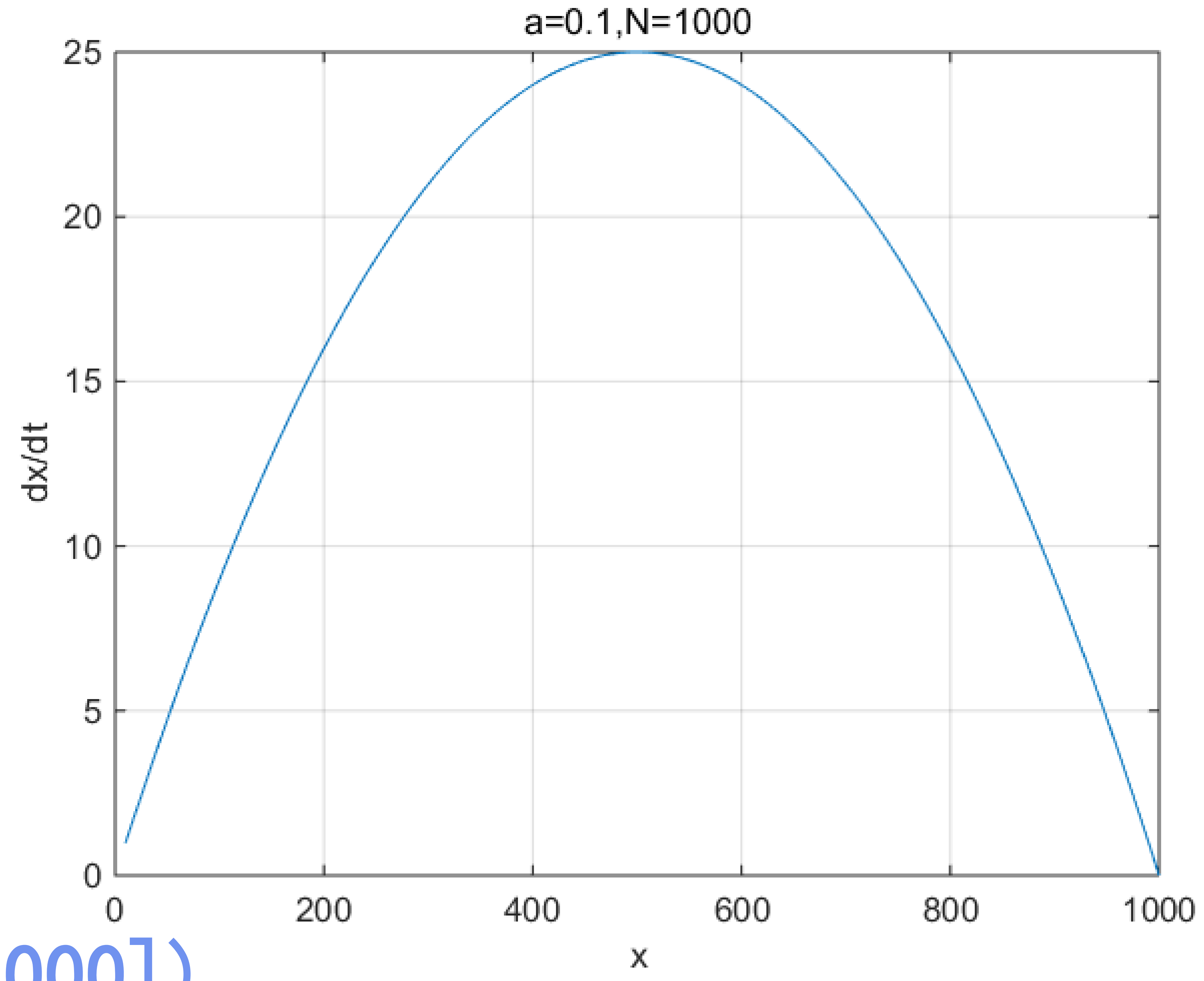
```
s =

N/(1+exp(-a*t)*C1*N)
```

$$x(t) = \frac{N}{1+CNe^{-at}}.$$

常微分方程

阻滞增长模型： $x'(t) = a(1 - \frac{x}{N}) \cdot x.$

观察 x' 随着 x变化的图形，随着种群总数的增加， 种群增速先是不断上升， 后来增速下降， 并在 $x = N$时停止增长(增速为0).




```
syms x
ezplot(0.1*(1-(x/1000))*x,[0,1000])
%当a=0.1, N=1000时的增长率图
```

增长率

常微分方程

阻滞增长模型：$x'(t) = a(1 - \frac{x}{N}) \cdot x.$

$$x(t) = \frac{N}{1+CNe^{-at}}.$$

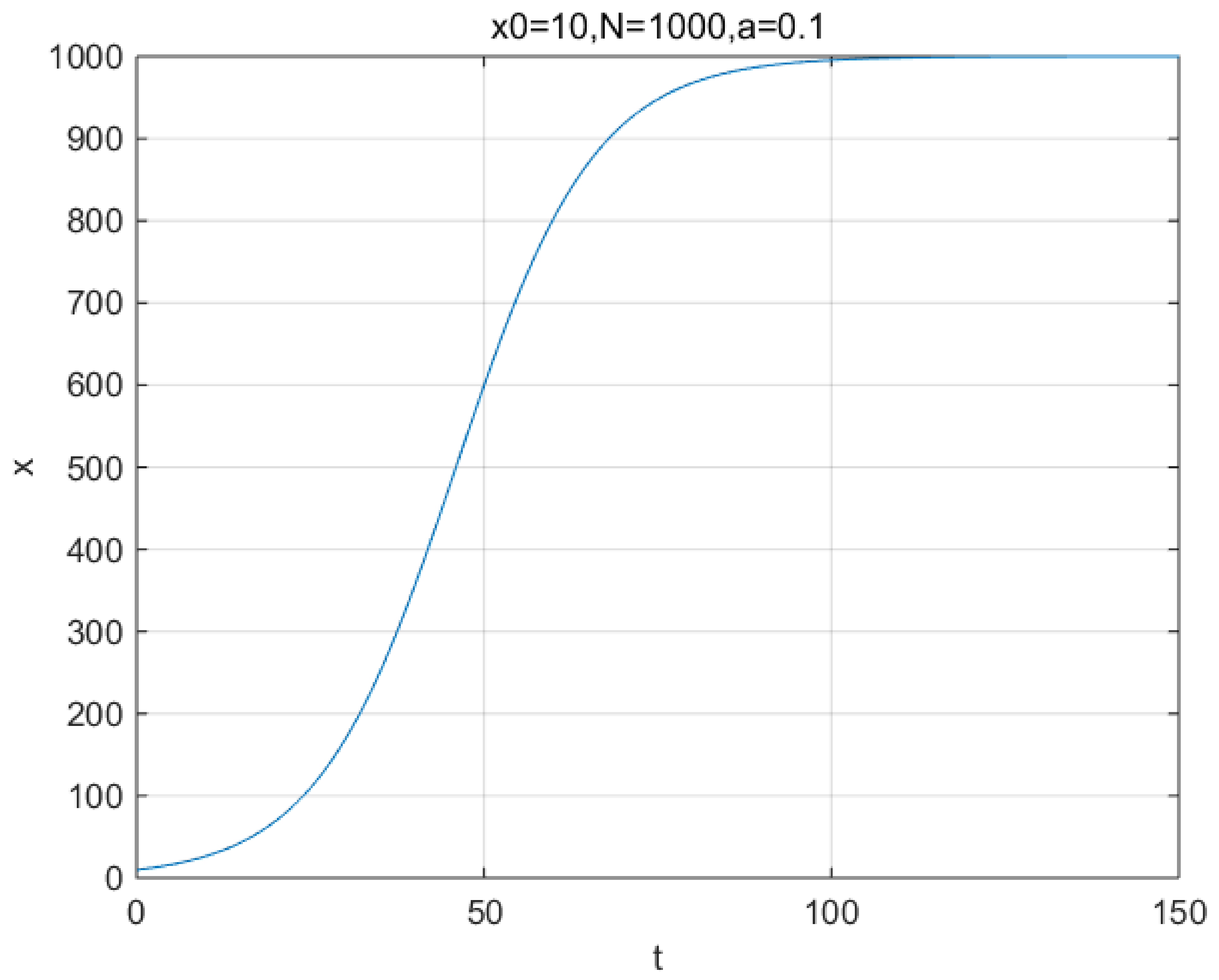
假设 $t=0$时刻种群总数为 x_0，带入 $x(t)$表达式可求出C，

这时解可以改写为：

$$x(t) = \frac{N}{1+(\frac{N}{x_0}-1)e^{-at}}.$$

随着时间推移，种群规模趋于环境允许的上限.

```
syms t
ezplot(1000/(1+(1000/10-1)*exp(-
0.1*t)),[0,150]
```



常微分方程

这里所介绍的常微分方程解法，都是求精确解，但是绝大多数常微分方程都是无法写出精确解的.

在实际应用中，经常以微分方程的数值解来代替精确解，另外还可以不去求解,转而用常微分方程定性理论来直接分析解的性态.

常微分方程的数值解

非线性常微分方程通常是无法求解析解的，既使是线性常微分方程，如果不能获得变系数的积分，也无法给出解的表达式.

因此，了解常微分方程的数值解法就显得非常重要了.

下面我们介绍几种求常微分方程数值解的方法.

常微分方程的数值解

求常微分方程（组）的数值解，就是选取自变量的若干个离散节点，求常微分方程的解在这些节点处的近似值.

我们知道，函数的导数是以差商的极限的形式定义的，在求常微分方程数值解时，一个很容易想到的思路就是反其道而行之，将导数转变为差商形式.

常微分方程的数值解

Euler法

对于带初值条件的常微分方程 $y'=f(t, y), y(t_0)=y_0$，首先选取适当的节点：$t_0<t_1<t_2<\cdots<t_n<t_{n+1}$.

通常是采用固定步长的方式，即令$t_{k+1}-t_k\equiv h$. 其次，在节点处用差商代替导数：

$$y'(t_k) \approx \frac{y(t_{k+1})-y(t_k)}{h} \Rightarrow y(t_{k+1}) \approx y(t_k)+h \cdot y'(t_k)$$

进一步：$y(t_{k+1}) \approx y(t_k)+h \cdot f(t_k, y(t_k))$.

Euler法的多种改进

在使用Euler法求解时，可以有多种不同的处理方法.

Euler法：$y(t_{k+1}) \approx y(t_k)+h \cdot f(t_k, y(t_k))$.

向后Euler法：$y(t_{k+1}) \approx y(t_k)+h \cdot f(t_{k+1}, \boxed{y(t_{k+1})})$.

以及改进Euler法：

$$\frac{y'(t_k)+y'(t_{k+1})}{2}$$

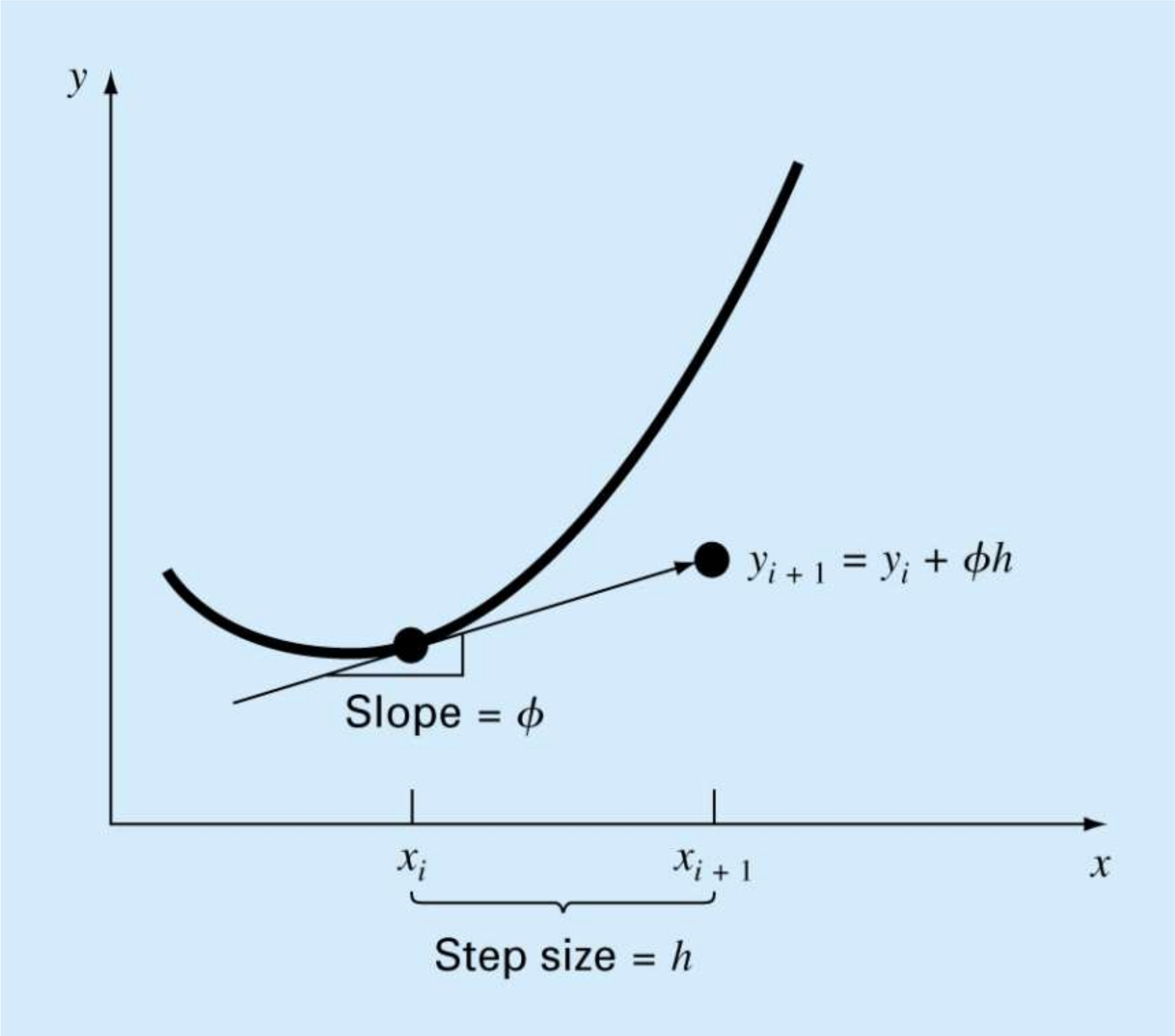
$$y(t_{k+1}) \approx y(t_k)+h\cdot \boxed{\frac{f(t_k, y(t_k))+f(t_{k+1}, y(t_{k+1}))}{2}}$$

$$= y(t_k)+h\cdot \frac{f(t_k, y(t_k))+f(t_{k+1}, y(t_{k+1})+h\cdot f(t_k, y(t_k)))}{2}$$

常微分方程的数值解

微分方程求数值解的基本思路(步进法):

后边节点值 = 前边节点值 + "斜率" × 步长



单步法一般形式：$y_{n+1} = y_n + \varphi_n h$

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) \\ y_0 = y(a) \end{cases}$$

$$\begin{cases} y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \\ y_0 = y(a) \end{cases}$$

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right] \\ y_0 = y(a) \end{cases}$$

常微分方程的数值解

例：用欧拉法求解初值问题

$$\frac{dy}{dx} = x - y,\quad 0 \le x \le 1, y(0) = 1$$

分别取h=0.075，0.0075，计算误差，画出精确解和数值解的图形。

精确解：

```
>> dsolve('Dy=x-y','y(0)=1','x')

ans =

x - 1+ 2*exp(-x)
```

常微分方程的数值解

```
function P=Euler(x0,y0,b,h)
%x0,y0--初值, b-x0取值的右端点, h—步长
n=(b-x0)/h; n=round(n);  %取整
X=zeros(n,1);Y=zeros(n,1);
k=1;
X(k)=x0;Y(k)=y0;
for k=1:n
    X(k+1)=X(k)+h;
    Y(k+1)=Y(k)+h*(X(k)-Y(k));%方程
    k=k+1;
end
y=X-1+2*exp(-X);%精确解
plot(X,Y,'mp',X,y,'b-')
Xlabel('自变量X'), ylabel('因变量y');
legend('精确解','数值解');
jwY=y-Y; %误差
xwY=jwY./y; %相对误差
k1=1:n;  k=[0,k1]; %序号
P=[k',X,Y,y,jwY,xwY]
```

Euler法:

$$y(t_{k+1}) \approx y(t_k) + h \cdot f(t_k, y(t_k)).$$

方程:

$$\frac{dy}{dx} = x - y,\quad 0 \le x \le 1,$$
$$y(0) = 1$$

精确解:

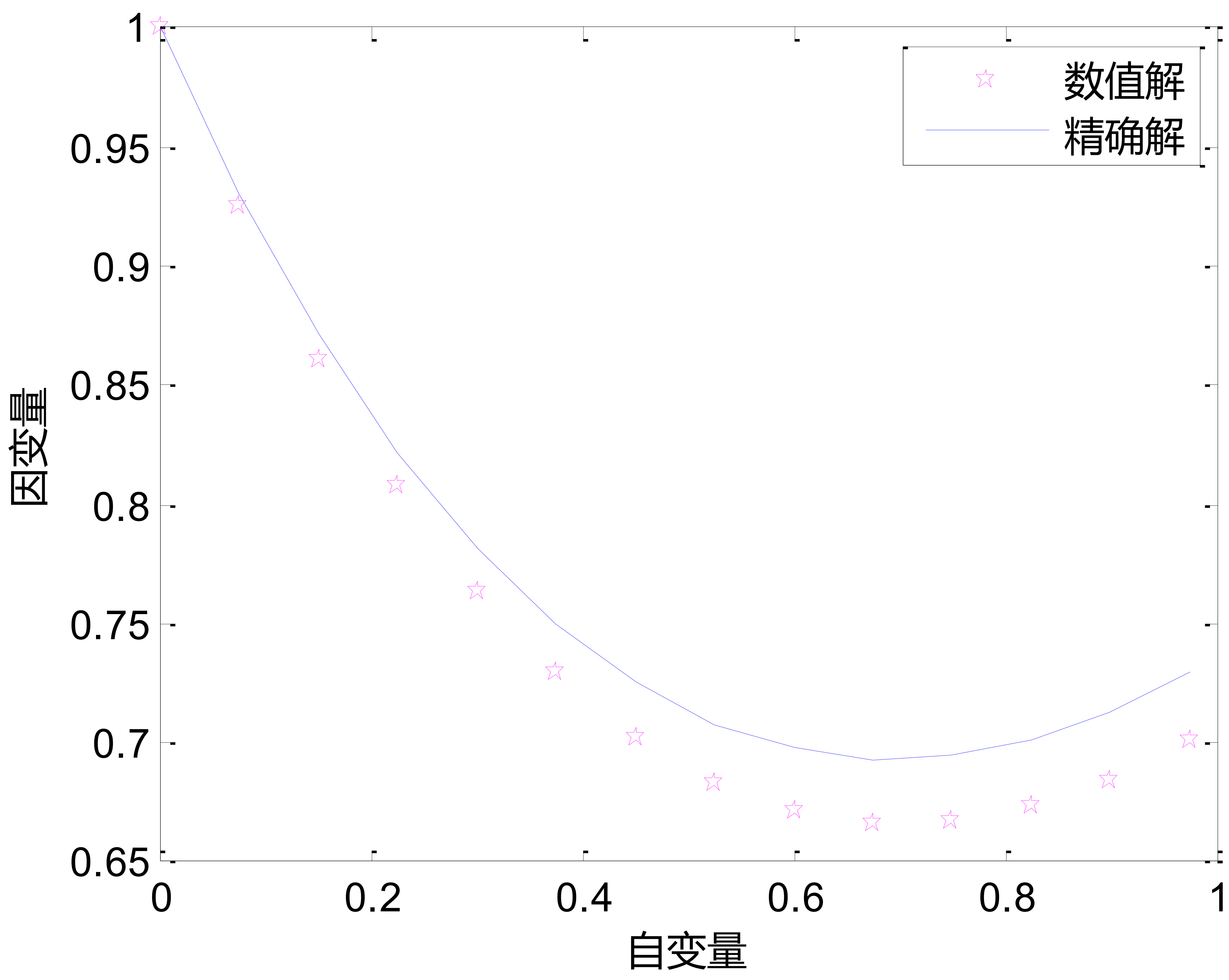
x + 2*exp(-x) - 1

常微分方程的数值解

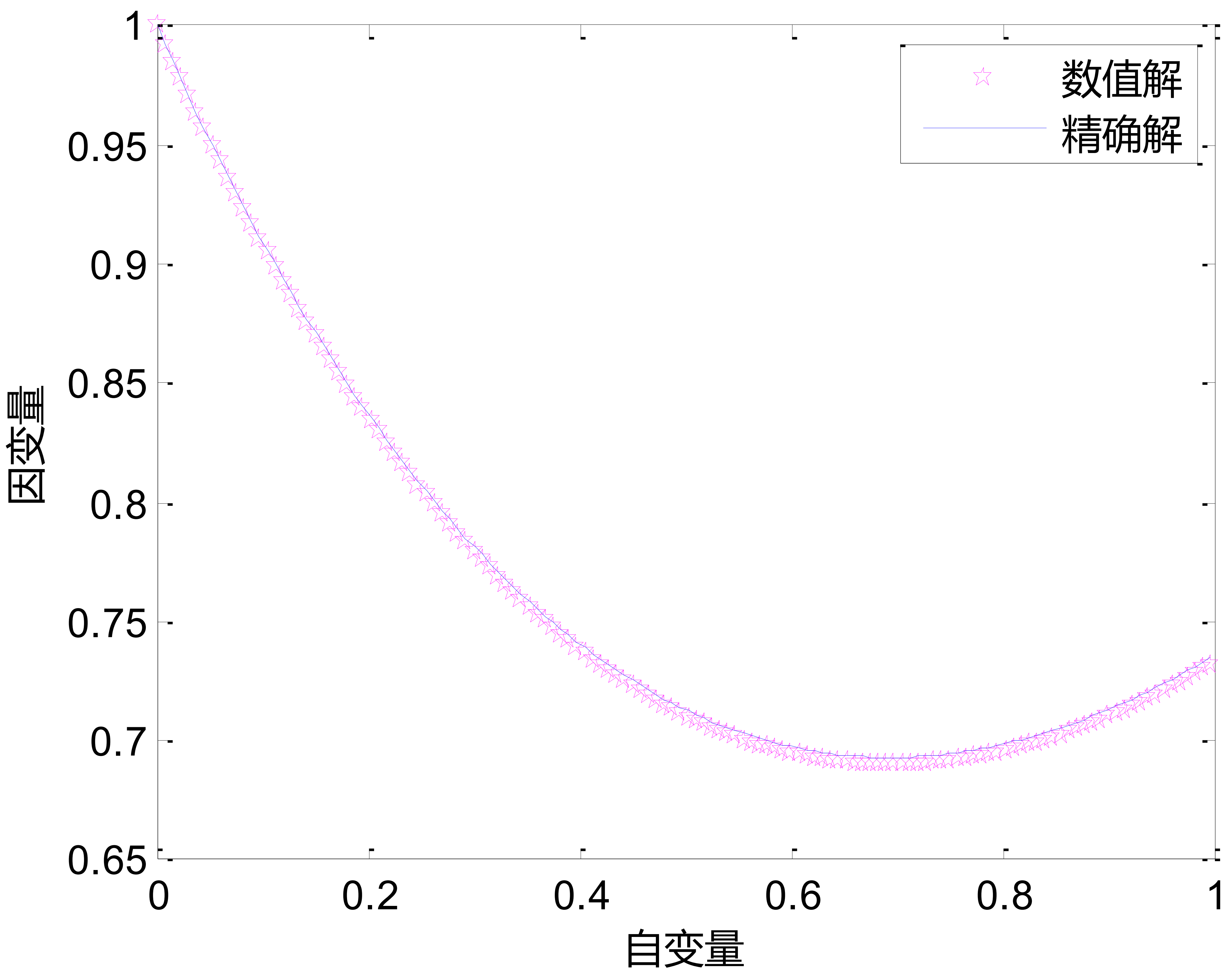
```
function P=Euler(x0,y0,b,h)
%x0,y0--初值, b-x0取值的右端点, h—步长
n=(b-x0)/h; n=round(n);  %取整
X=zeros(n,1);Y=zeros(n,1);
k=1;
X(k)=x0;Y(k)=y0;
for k=1:n
    X(k+1)=X(k)+h;
    Y(k+1)=Y(k)+h*(X(k)-Y(k));%方程
    k=k+1;
end
y=X-1+2*exp(-X);%精确解
plot(X,Y,'mp',X,y,'b-')
Xlabel('自变量X'), ylabel('因变量y');
legend('精确解','数值解');
jwY=y-Y; %误差
xwY=jwY./y; %相对误差
k1=1:n;  k=[0,k1]; %序号
P=[k',X,Y,y,jwY,xwY]
```

```
>> x0=0;
y0=1;
b=1;
h=0.075;
P=Euler(x0,y0,b,h)

P =

         0         0    1.0000    1.0000         0         0
    1.0000    0.0750    0.9250    0.9305    0.0055    0.0059
    2.0000    0.1500    0.8613    0.8714    0.0102    0.0117
    3.0000    0.2250    0.8079    0.8220    0.0141    0.0172
    4.0000    0.3000    0.7642    0.7816    0.0174    0.0223
    5.0000    0.3750    0.7294    0.7496    0.0202    0.0270
    6.0000    0.4500    0.7028    0.7253    0.0225    0.0310
    7.0000    0.5250    0.6838    0.7081    0.0243    0.0343
    8.0000    0.6000    0.6719    0.6976    0.0257    0.0368
    9.0000    0.6750    0.6665    0.6933    0.0268    0.0386
   10.0000    0.7500    0.6672    0.6947    0.0276    0.0397
   11.0000    0.8250    0.6734    0.7015    0.0281    0.0400
   12.0000    0.9000    0.6847    0.7131    0.0284    0.0398
   13.0000    0.9750    0.7009    0.7294    0.0285    0.0391
```

常微分方程的数值解



h=0. 075



h=0. 0075

常微分方程的数值解

龙格-库塔方法

实际应用中发现，Euler法的精度无法满足要求，同时综合考虑多个节点处的斜率，可以有效提高精度.

龙格-库塔方法就是一种考虑多个节点的方法，适用面比较广、精度也较高.

具体的思路如下：

$$y_{k+1} \approx y_k + \sum_{i=1}^{N} c_i K_i.$$

其中 $K_1 = f(x_k, y_k)$， $K_i = f(x_k + p_i h,\ y_k + \sum_{j=1}^{i-1} b_{ij} K_j)$，

h为步长.

Runge—Kutta 法的基本思想

前面公式共性：以函数 $f(x,y)$在若干点上函数值的线性组合代替φ．
公式关键是系数与点的选取

单步法一般形式：$y_{n+1}=y_n+\varphi(x_n,y_n,x_{n+1},y_{n+1},h)\,h$

R-K构造思想：

使近似公式在(x_n,y_n)处的Taylor 展开与解 $y(x)$在 x_n 的Taylor 展开式的前 $p+1$项重合,从而得到 p阶公式.

Runge-Kutta: $\varphi=c_1K_1+c_2K_2+\cdots+c_pK_p$

RK方法的构造　RK公式的一般形式为

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

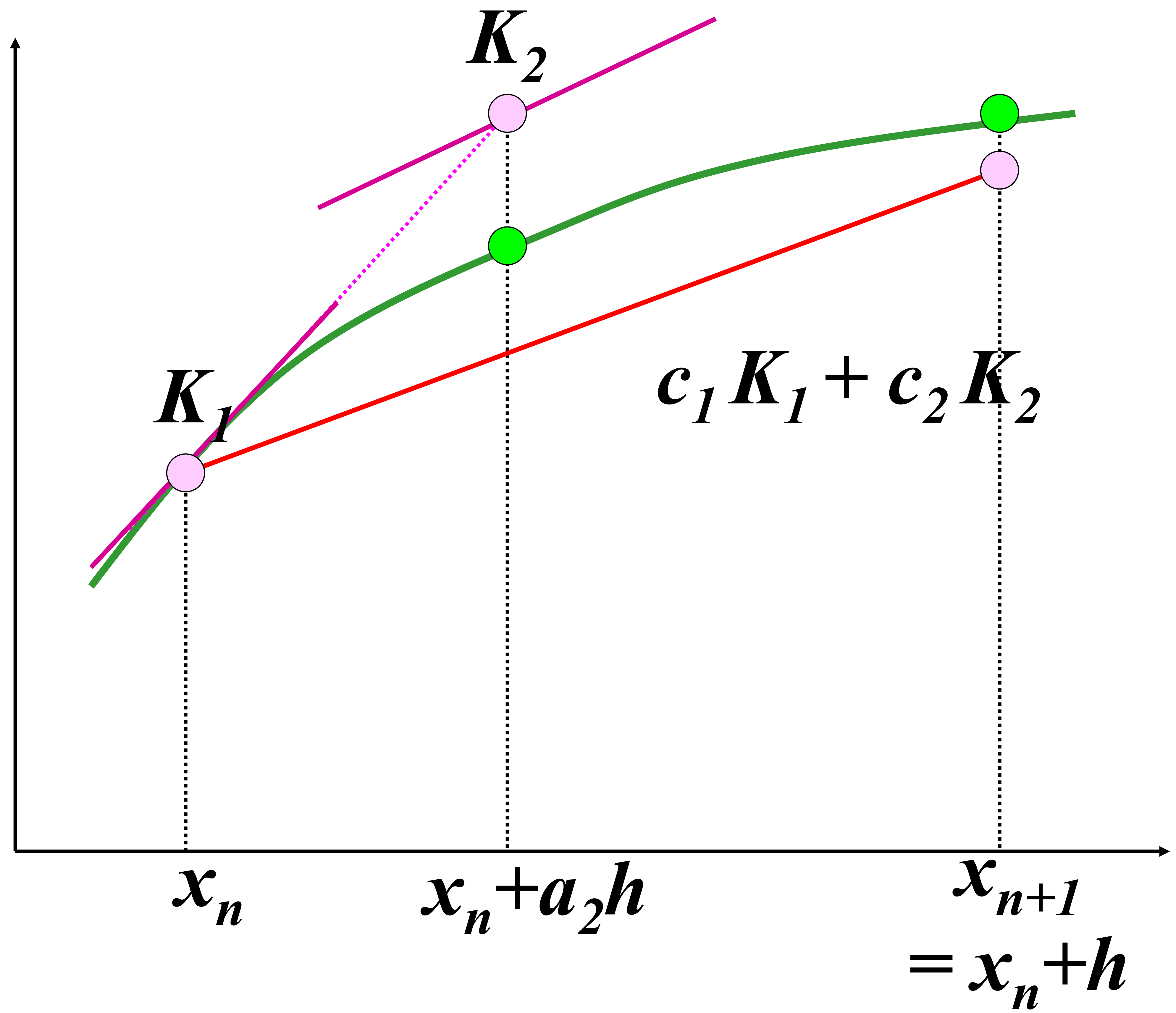
$$\varphi(x_n, y_n, h) = c_1K_1 + c_2K_2 + \cdots + c_pK_p$$

$$\begin{cases} K_1 = f(x_n, y_n) \\ K_2 = f(x_n + a_2h, y_n + b_{21}K_1h) \\ K_3 = f(x_n + a_3h, y_n + b_{31}K_1h + b_{32}K_2h) \\ \quad\vdots \\ K_p = f(x_n + a_ph, y_n + b_{p,1}K_1h + b_{p,2}K_2h + \cdots + b_{p,p-1}K_{p-1}h) \end{cases}$$

其中 a_i, b_{ij}, c_i 为待定系数.

以 $p=2$为例，RK公式

$$\begin{cases} y_{n+1}=y_n+h(c_1K_1+c_2K_2) \\ K_1=f(x_n,y_n) \\ K_2=f(x_n+a_2h,y_n+hb_{21}K_1) \end{cases}$$



几何意义：

“加权平均”斜率

$p=2$时RK公式的推导：
$$\begin{cases} y_{n+1}=y_n+h(c_1K_1+c_2K_2) \\ K_1=f(x_n,y_n) \\ K_2=f(x_n+a_2h,y_n+hb_{21}K_1) \end{cases}$$

$$y_{n+1}=y_n+h\left[c_1f(x_n,y_n)+c_2f\left(x_n+a_2h,y_n+hb_{21}f(x_n,y_n)\right)\right]$$

$$=y_n+h\left\{c_1f(x_n,y_n)+c_2\left[f(x_n,y_n)+a_2hf_x'(x_n,y_n)+b_{21}hf(x_n,y_n)f_y'(x_n,y_n)+O(h^2)\right]\right\}$$

$$=y_n+(c_1+c_2)hf(x_n,y_n)+c_2h^2\left[a_2f_x'(x_n,y_n)+b_{21}f(x_n,y_n)f_y'(x_n,y_n)\right]+O(h^3)$$

常微分方程的数值解

$$y_{n+1}=y_n+(c_1+c_2)hf(x_n,y_n)+c_2h^2\left[a_2f'_x(x_n,y_n)+b_{21}f(x_n,y_n)f'_y(x_n,y_n)\right]+O(h^3)$$

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

$$=y_n+hf(x_n,y_n)+\frac{h^2}{2}\left[f'_x(x_n,y_n)+f'_y(x_n,y_n)f(x_n,y_n)\right]+O(h^3)$$

为使 $y(x_{n+1})-\overline{y}_{n+1}=O(h^3)$ $\Rightarrow$ $\begin{cases} c_1+c_2=1 \\ c_2a_2=\dfrac{1}{2} \\ c_2b_{21}=\dfrac{1}{2} \end{cases}$

$$\begin{cases} c_1 + c_2 = 1 \\ c_2 a_2 = \dfrac{1}{2} \\ c_2 b_{21} = \dfrac{1}{2} \end{cases}$$

特别地，令 $c_2 = \frac{1}{2}$，则 $c_1 = \frac{1}{2}, a_2 = b_{21} = 1$

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

这就是改进Euler公式，二阶方法。

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1) \end{cases}$$

$$\begin{cases} y_p = y_n + hf(x_n, y_n) \\ y_q = y_n + hf(x_{n+1}, y_p) \\ y_{n+1} = \dfrac{1}{2}(y_p + y_q) \end{cases}$$

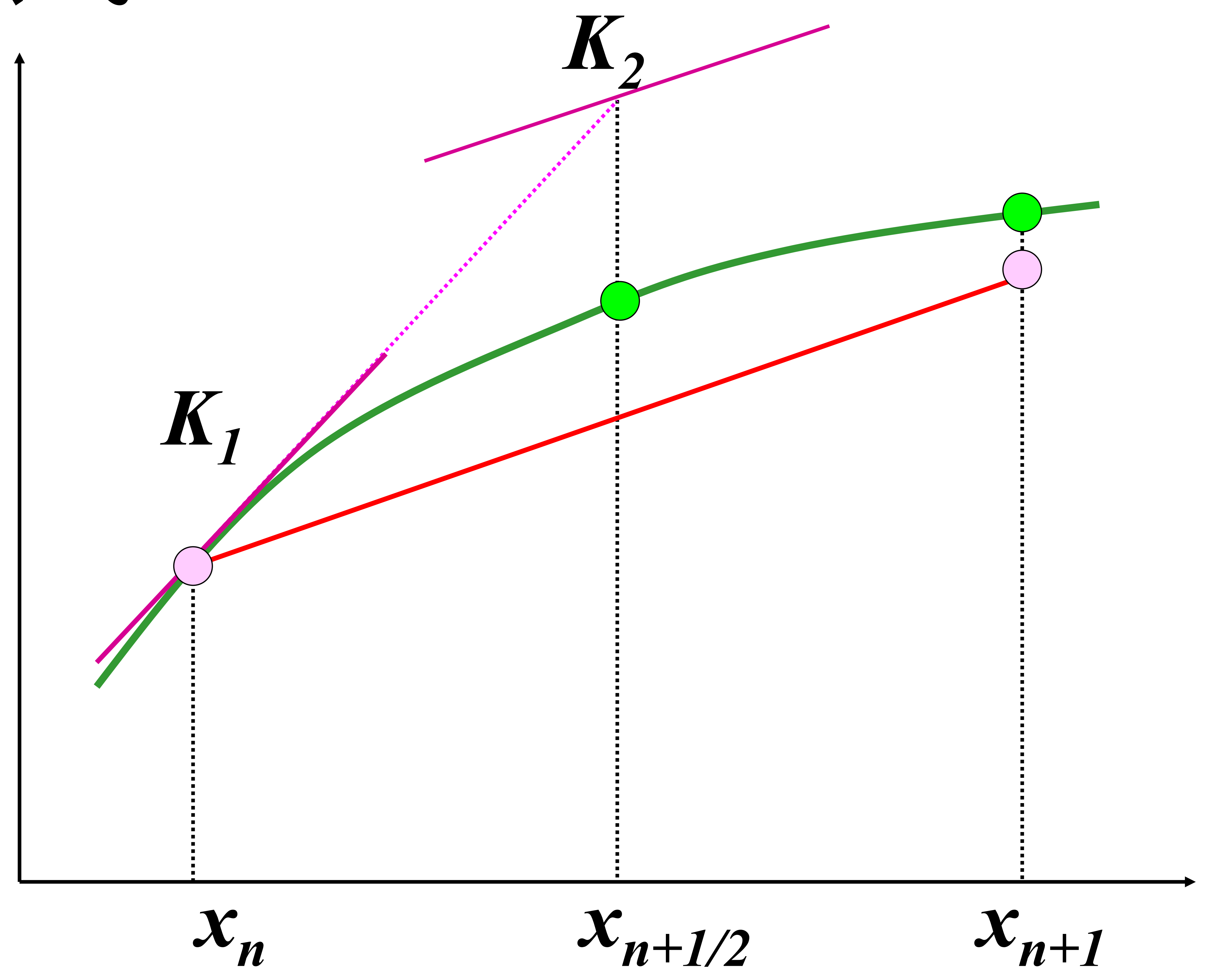
常微分方程的数值解

$$\begin{cases} c_1 + c_2 = 1 \\ c_2 a_2 = \dfrac{1}{2} \\ c_2 b_{21} = \dfrac{1}{2} \end{cases}$$

令$c_2 = 1$,则 $c_1 = 0, a_2 = b_{21} = \dfrac{1}{2}$,

则得如下中点公式

$$\begin{cases} y_{n+1} = y_n + hK_2 \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1) \end{cases}$$



三阶 Runge-Kutta 公式的一般形式：

$$\phi = c_1K_1 + c_2K_2 + c_3K_3$$

$$\begin{cases} K_1 = f(x_n, y_n) \\ K_2 = f(x_n + a_2h, y_n + b_{21}K_1h) \\ K_3 = f(x_n + a_3h, y_n + b_{31}K_1h + b_{32}K_2h) \\ y_{n+1} = y_n + (c_1K_1 + c_2K_2 + c_3K_3)h \end{cases}$$

常微分方程的数值解

$$y_{n+1}=y_n+\frac{1}{8}(2K_1+3K_2+3K_3)h$$

$$\begin{cases}K_1=f(x_n,y_n)\\ K_2=f(x_n+\frac{2}{3}h,y_n+\frac{2}{3}K_1h)\\ K_3=f(x_n+\frac{2}{3}h,y_n+\frac{2}{3}K_2h)\end{cases}$$

$$y_{n+1}=y_n+\frac{1}{9}(2K_1+3K_2+4K_3)h$$

$$\begin{cases}K_1=f(x_n,y_n)\\ K_2=f(x_n+\frac{1}{2}h,y_n+\frac{1}{2}K_1h)\\ K_3=f(x_n+\frac{3}{4}h,y_n+\frac{3}{4}K_2h)\end{cases}$$

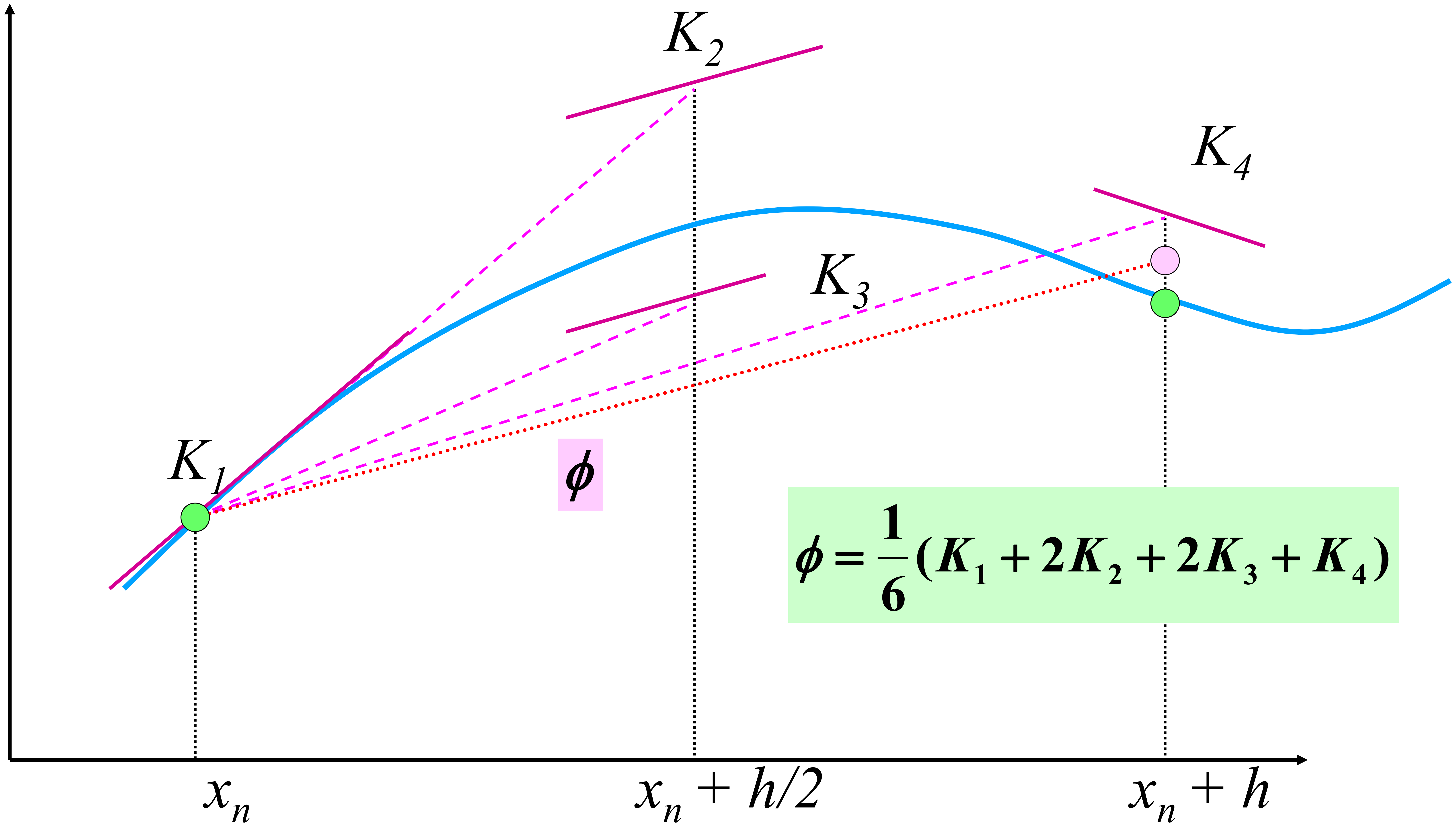
$$\begin{cases}y_{n+1}=y_n+\frac{h}{6}(K_1+4K_2+K_3)\\ K_1=f(x_n,y_n)\\ K_2=f(x_n+\frac{h}{2},y_n+\frac{h}{2}K_1)\\ K_3=f(x_n+h,y_n-hK_1+2hK_2)\end{cases}$$

经典四阶RK公式

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

$$\begin{cases} K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1) \\ K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2) \\ K_4 = f(x_n + h, y_n + hK_3) \end{cases}$$

经典四阶RK公式的几何意义



注意：当 $p \le 4$时， RK公式的最高阶数为 p;

当 $p > 4$时， RK公式的最高阶数小于 p.

常微分方程的数值解

例：用二阶龙格-库塔方法求解初值问题

$$\frac{dy}{dx} = 1 - \frac{2xy}{1+x^2},\quad 0 \le x \le 2,\quad y(0) = 0$$

取h=1/4 和 0.0075，画出精确解和数值解的图形。

精确解：

```
dsolve('Dy=1-2*x*y/(1+x^2)','y(0)=0','x')

ans =

1/3/(1+x^2)*x^3+x/(1+x^2)
```

常微分方程的数值解

```
function P=Runge_Kutta_2(xmin,xmax,y0,h)
%xmin,xmax—x取值的最小值和最大值
%y0—初值  %h—迭代步长
x=xmin:h:xmax;
n=length(x);  %n=size(x);  %x的行数和列数
y=zeros(1,n);
y(1)=y0;
for i=1:n-1
    k1=1-2*x(i)*y(i)/(1+x(i)^2);
    k2=1-2*(x(i)+h)*(y(i)+h*k1)/(1+(x(i)+h)^2);
    y(i+1)=y(i)+(h/2)*(k1+k2);
end
y1=(x+1/3*x.^3)./(1+x.^2);   %精确解
plot(x,y,'mp',x,y1,'b-');
k=1:n;
P=[k',x',y',y1']
```

$$y_{k+1} \approx y_k + \frac{h}{2}(K_1 + K_2)$$

$$K_1 = f(x_k, y_k),$$

$$K_2 = f(x_k + h,\ \ y_k + hK_1)$$

$$\frac{dy}{dx} = 1 - \frac{2xy}{1+x^2},\ \ 0 \le x \le 2,$$

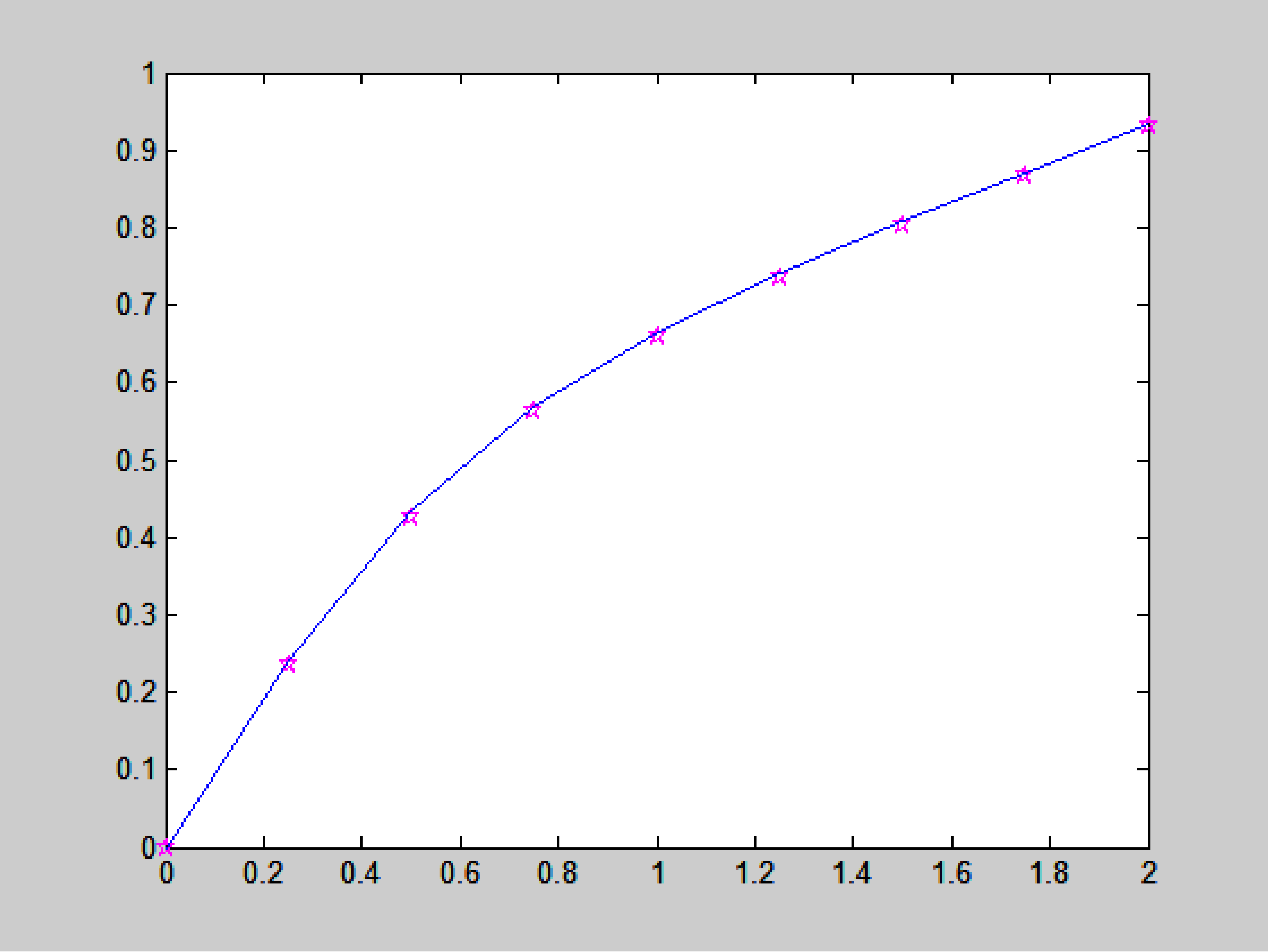
$$y(0) = 0$$

精确解：

(x*(x^2 + 3))/(3*(x^2 + 1))

常微分方程的数值解

```
xmin=0;xmax=2;h=1/4;y0=0;
P=Runge_Kutta_2(xmin,xmax,y0,h)

P =

    1.0000         0         0         0
    2.0000    0.2500    0.2353    0.2402
    3.0000    0.5000    0.4257    0.4333
    4.0000    0.7500    0.5623    0.5700
    5.0000    1.0000    0.6601    0.6667
    6.0000    1.2500    0.7367    0.7419
    7.0000    1.5000    0.8038    0.8077
    8.0000    1.7500    0.8675    0.8705
    9.0000    2.0000    0.9310    0.9333
```



求常微分方程数值解的函数

求常微分方程的数值解有多种算法，因此可供使用的函数也有多个. 常用的函数有：

函数名	简介	适用对象
ode45	单步，4/5阶龙格-库塔法	大部分ODE
ode23	单步，2/3阶龙格-库塔法	快速、精度不高的求解
ode113	多步，Adams算法	误差要求严格或计算复杂

注：上述函数仅适用于非刚性（nonstiff）方程（组），即其特征值的实部绝对值差异比较小.

常微分方程的数值解

所谓刚性方程（组），其数值解只有在步长很小时才会稳定，步长较大时解就会不稳定. 在具体应用中，如果使用常用函数长时间无结果，可以考虑换用如下函数：

函数名	简介	适用对象
ode23t	采用梯形算法	具有一定的刚性特点
ode15s	多步，反向数值积分法	ode45失效时可以试用
ode23s	单步，2阶Rosebrock算法	精度设定较低时，速度快
ode23tb	采用梯形算法	精度设定较低时，速度快

常微分方程的数值解

例： 求解初值问题

$$\frac{dy}{dx} = \frac{2x}{3y^2}, \quad 0 \le x \le 1.2, \qquad y(0) = 1$$

精确到 0.0001，画出精确解和数值解的图形。

精确解：

```
dsolve('Dy=2*x/(3*y^2)','y(0)=1','x')
```

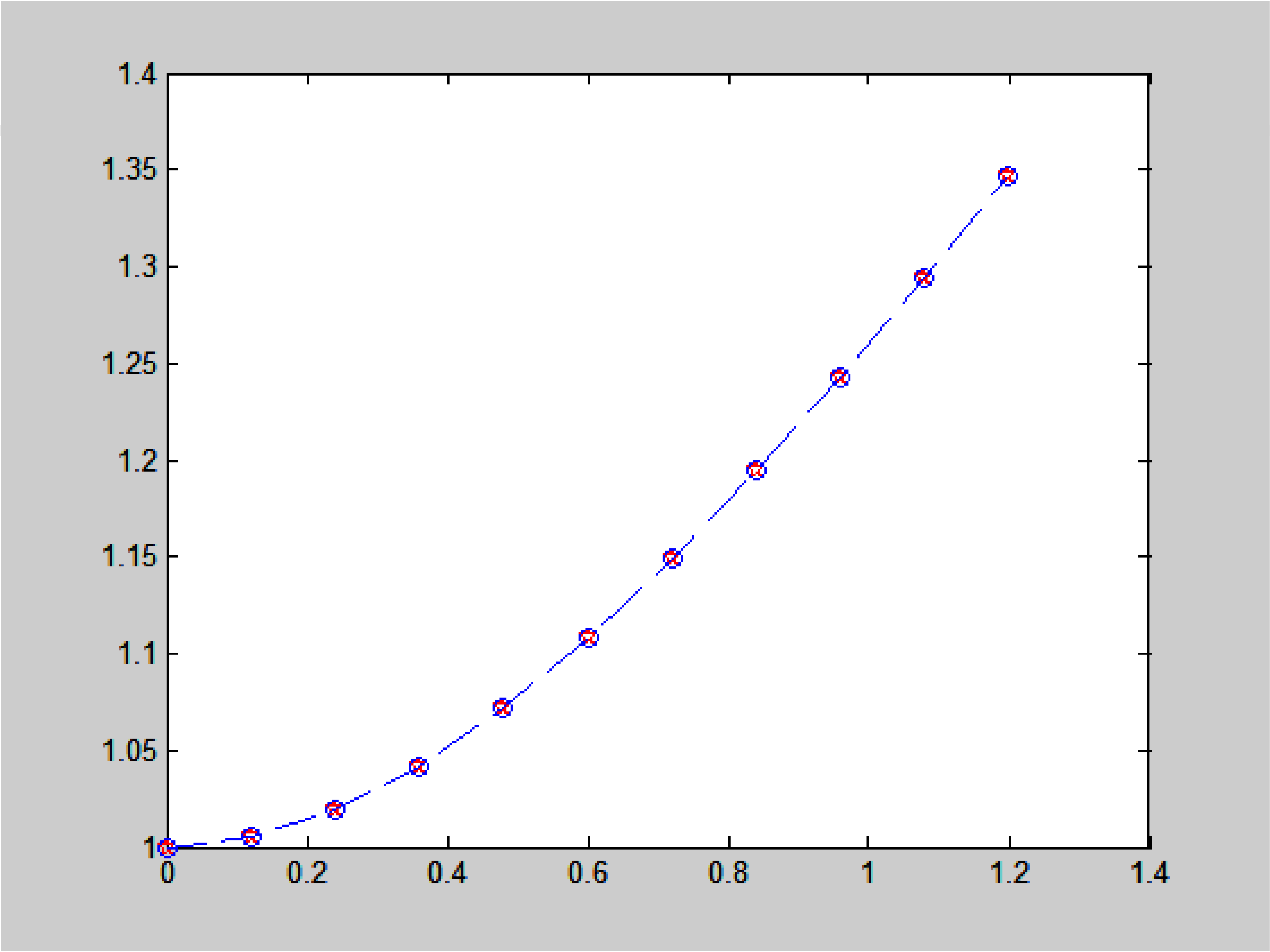
```
ans =

(1+x^2)^(1/3)
```

常微分方程的数值解

➢ 建立函数文件：

```
function y1=funfcn(x,y)
y1=2*x/(3*y^2);
```

➢ 比较ode23和精确解：



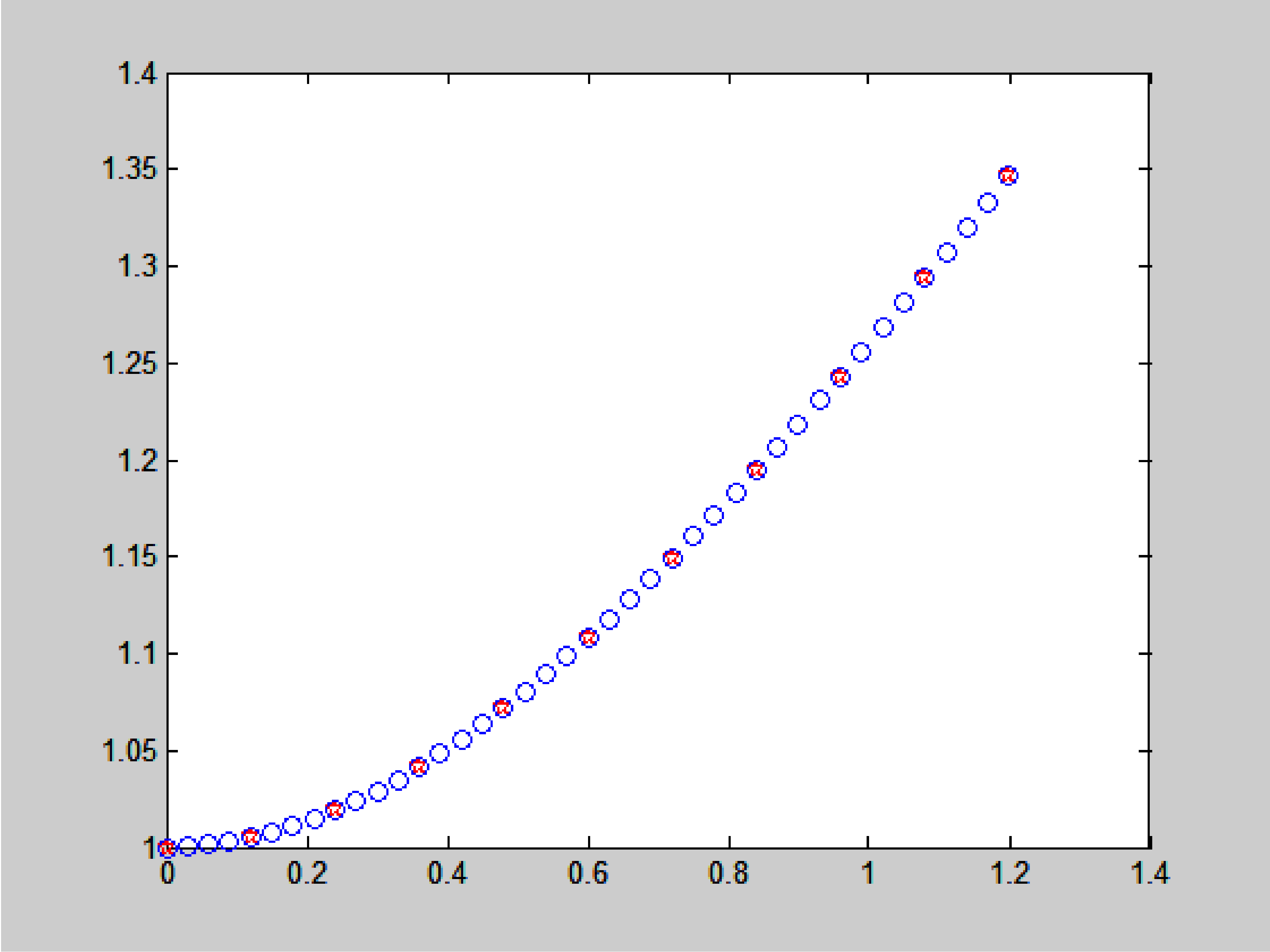

```
options=odeset('RelTol',1e-4,'AbsTol',1e-4);
[t,y]=ode23(@funfcn,[0,1.2],1,options);   %t, y是列向量
yf=(t.^2+1).^(1/3);   %精确解
plot(t,y(:,1),'rp--',t,yf,'bo--')
```

```
options=odeset('reltol',rt,'abstol',at)
[t,y]= solver('odefun',tspan,y0,options)
```

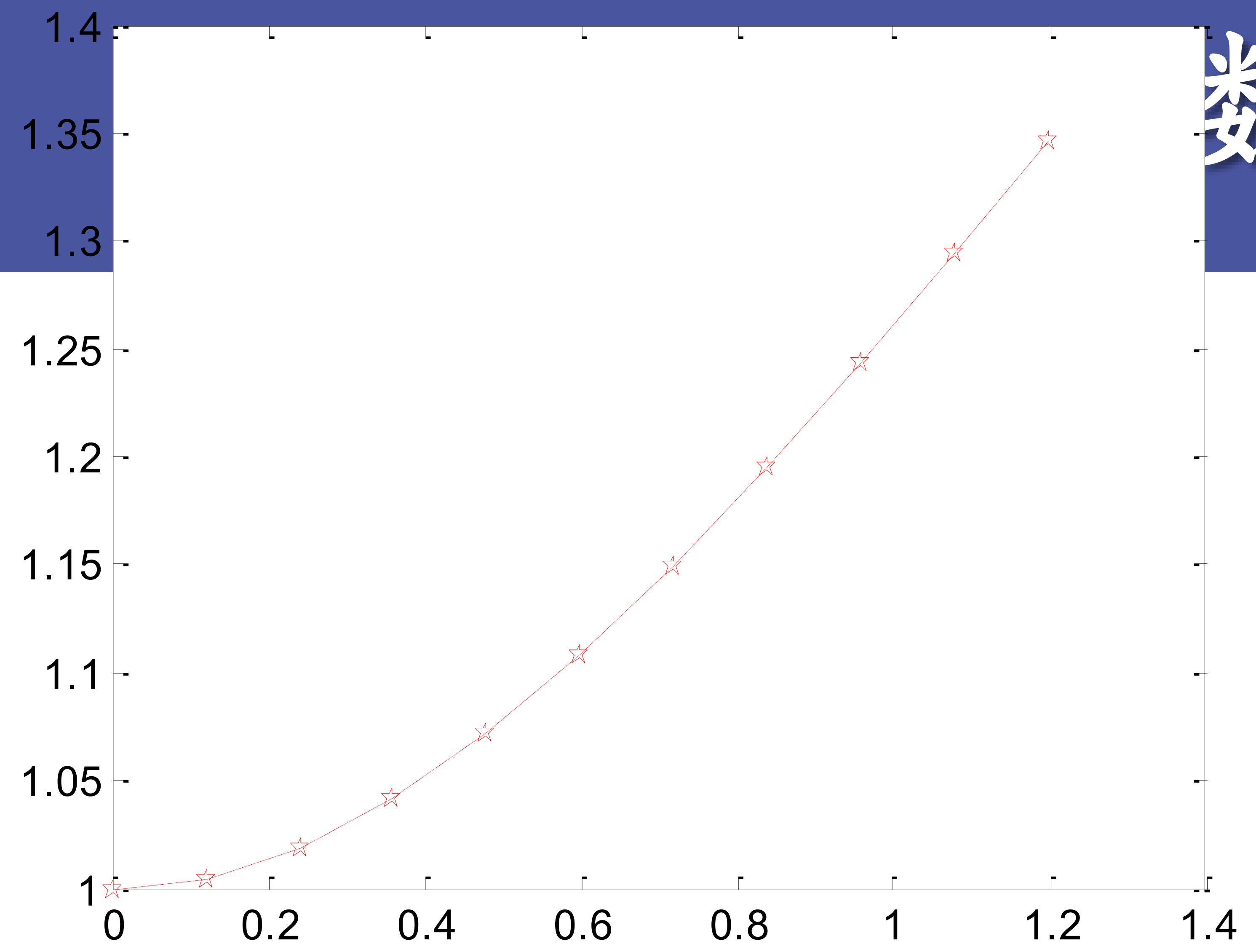
常微分方程的数值解

➢ 比较ode23和ode45的数值解：

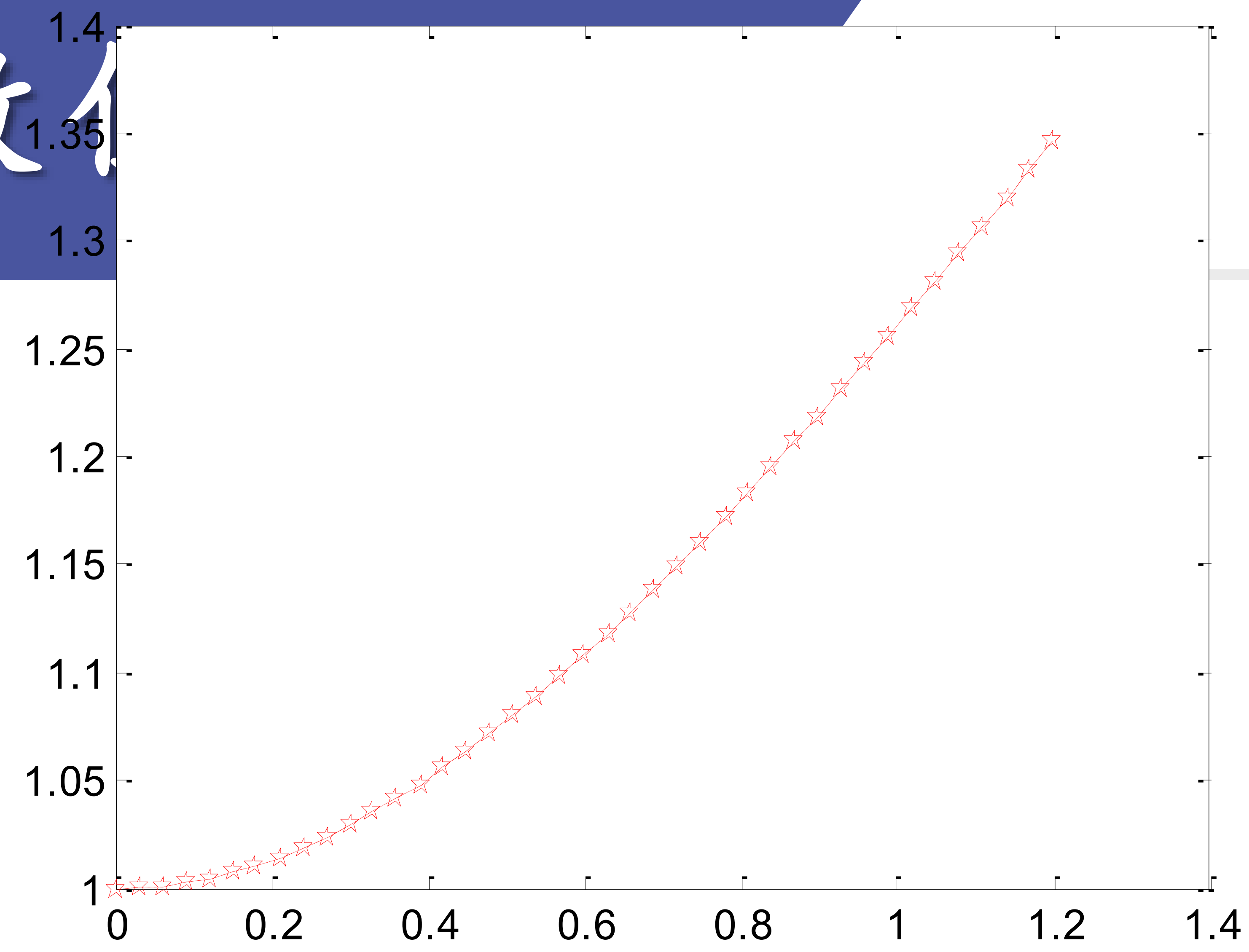
```
options=odeset('RelTol',1e-4,'AbsTol',1e-4);
[t,y]=ode23(@funfcn,[0,1.2],1,options);
[t1,y1]=ode45(@funfcn,[0,1.2],1,options);   %取点更密
plot(t,y(:,1),'rp',t1,y1(:,1),'bo')
```



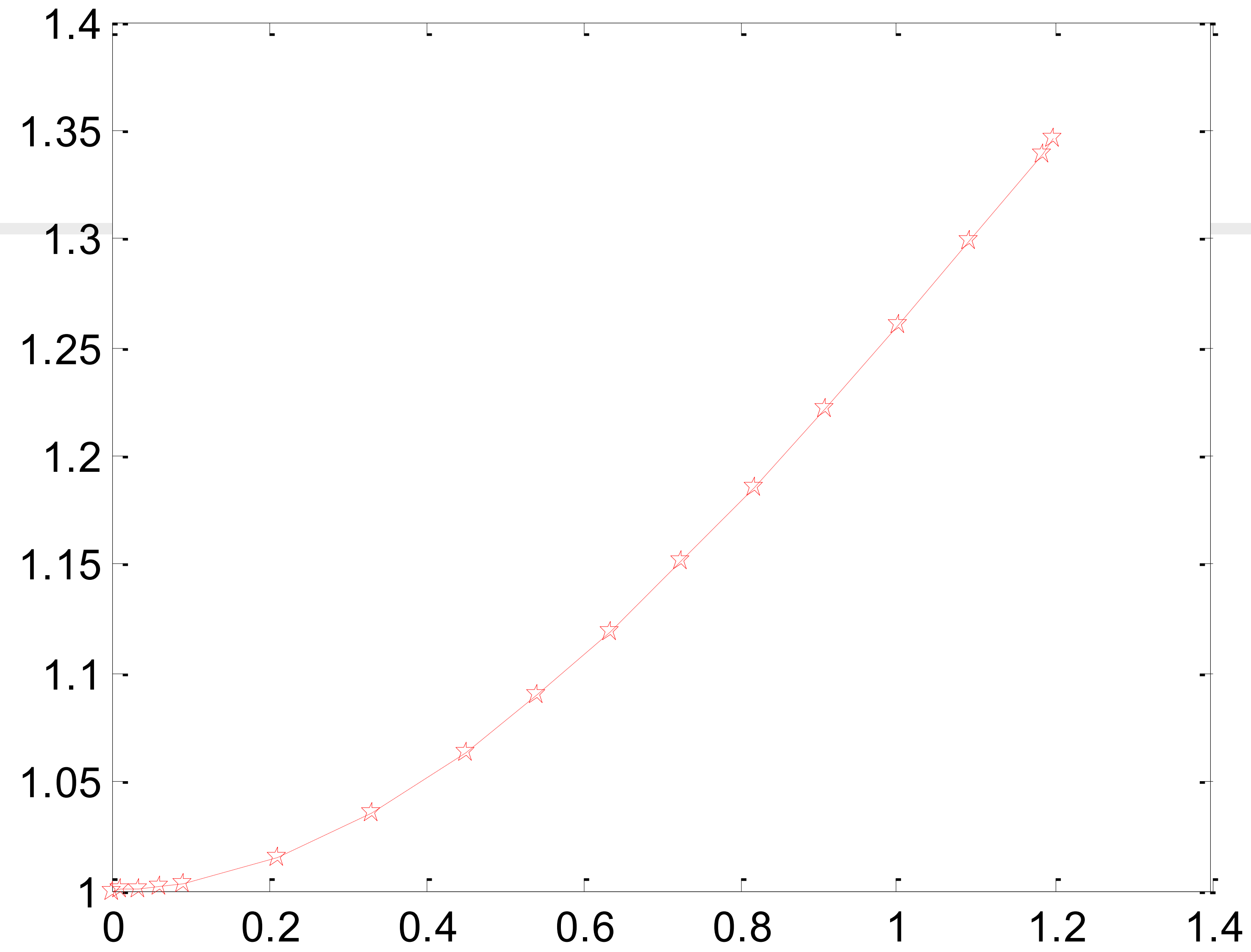
数



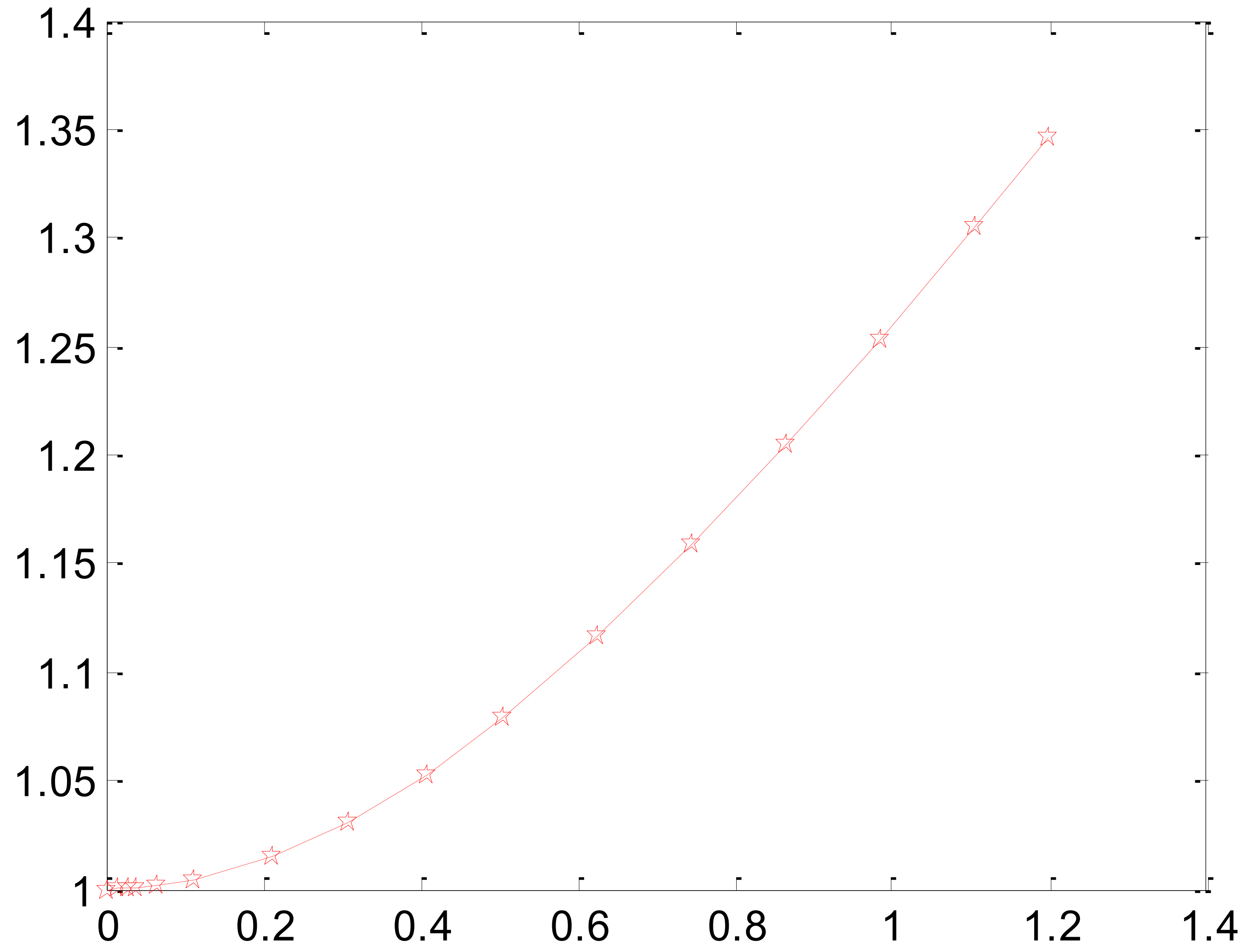
ode23



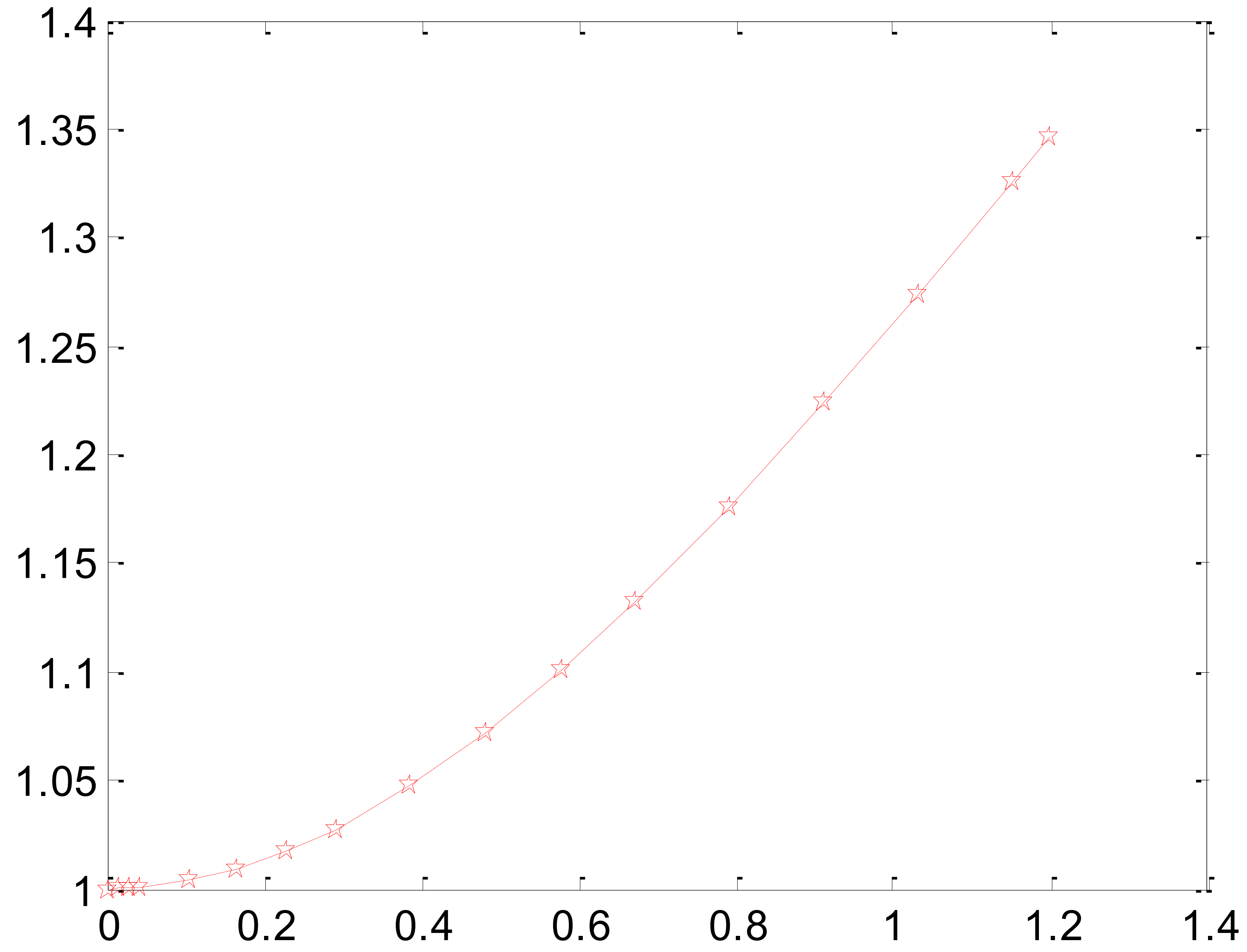
ode45



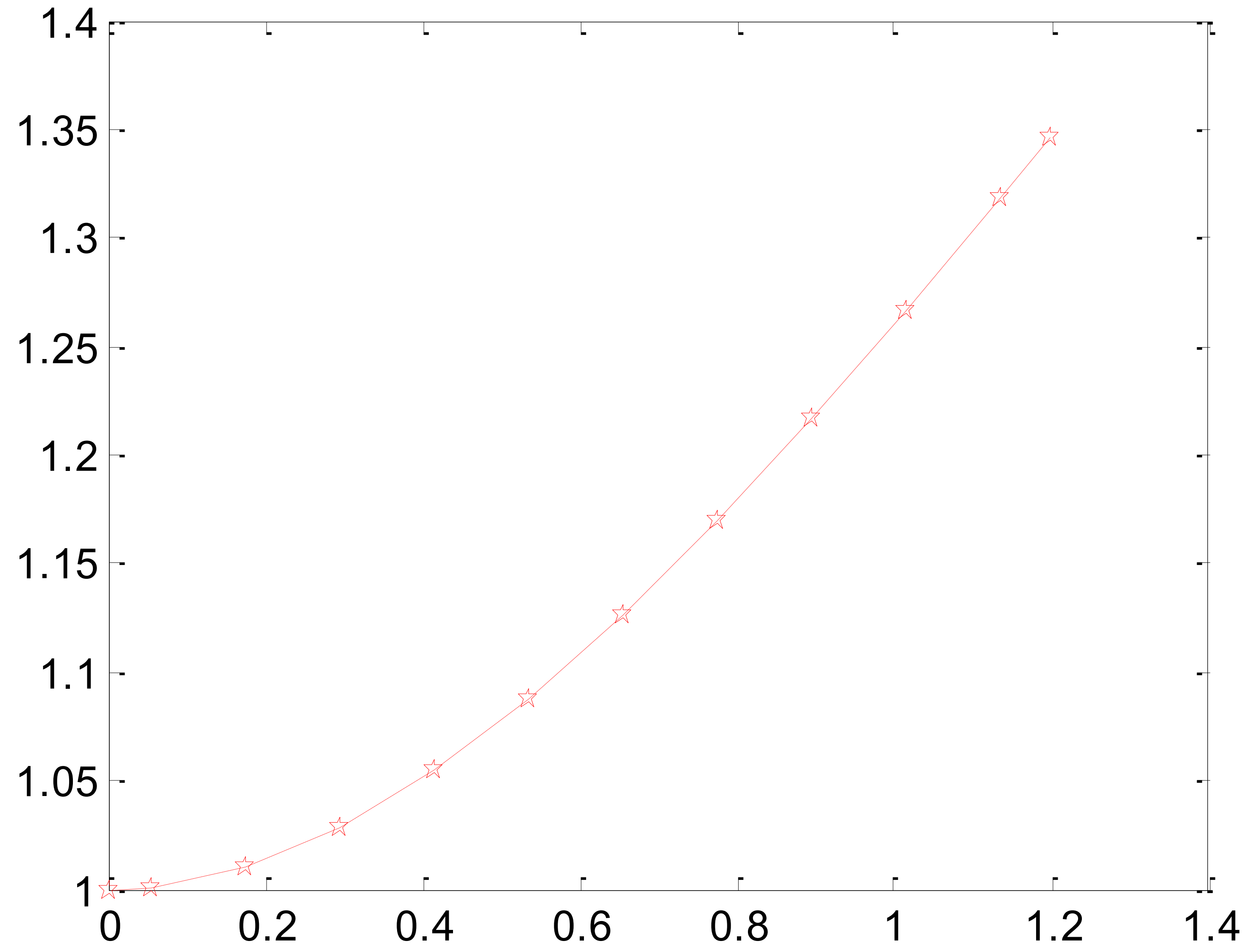
ode23t



ode113



ode15s



ode23tb

常微分方程的数值解

常微分方程数值求解的命令

求常微分方程的数值解，Matlab中的命令格式为：

[t, y]= solver ('odefun', tspan, y0, options)

solver—选择ode45等函数名；

odefun—待解方程或方程组的m文件名；

tspan—自变量的区间$[t_0, t_f]$，t_0为初始点；

y0—初始值；

options—用于设定误差限制. 命令格式为：

options=odeset('reltol', rt, 'abstol', at)

rt—输入相对误差，at—输入绝对误差.

常微分方程的数值解

例1. 人口指数增长模型

假设任意时刻，人口的增速与人口总数$x(t)$成正比. 建立常微分方程模型：$x'(t)=kx(t)$, $x(0)=x0$.

该方程的初值解为：$x(t)=\mathrm{x0}\cdot e^{kt}$. 下面求其数值解.

设$k=0.05$, $x0=1000$. 首先建立描述方程的m文件.

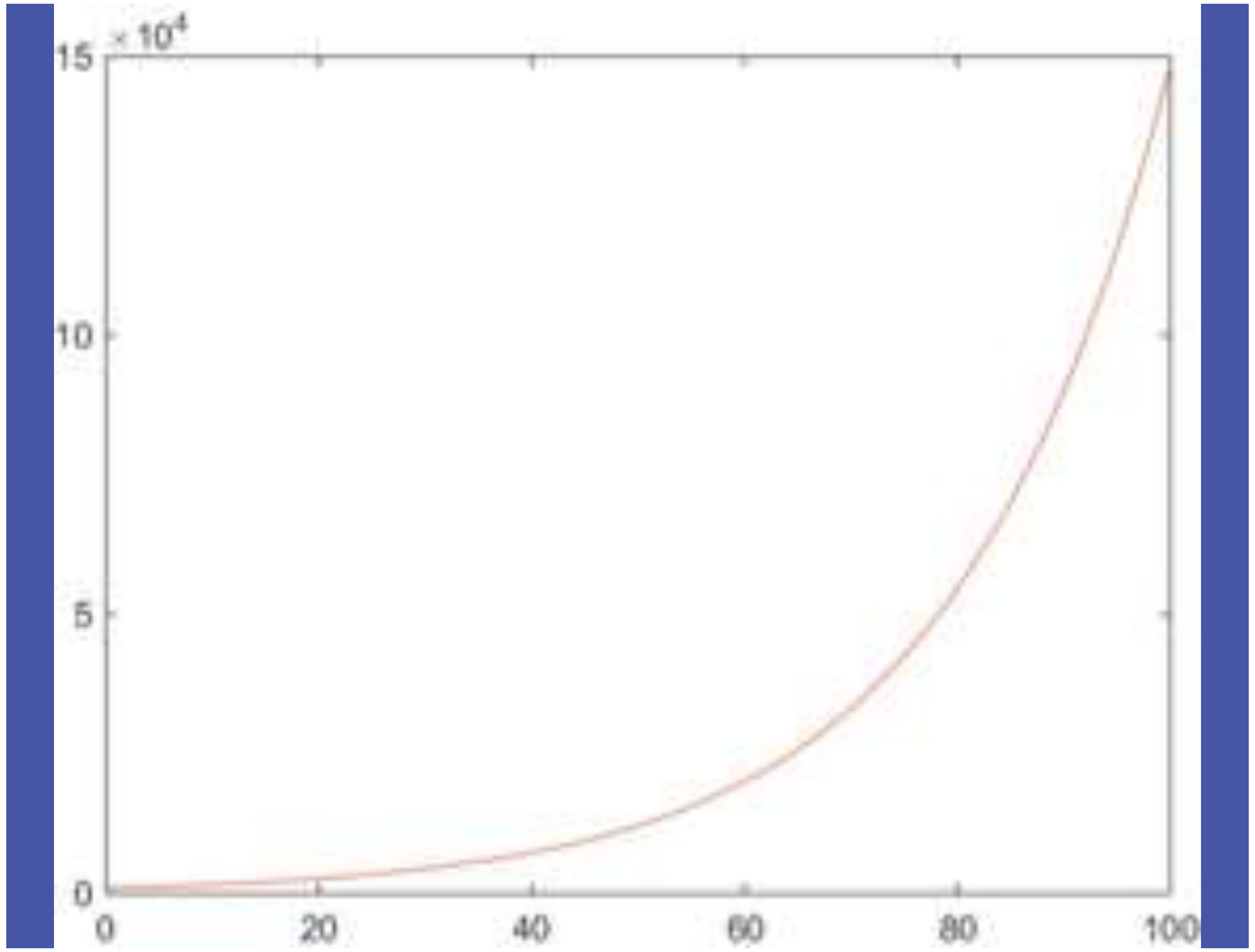
```
function dx=human(t, x)
dx=0.05*x
```

并保存为human.m.

常微分方程的数值解

接着使用命令：

```
>> [t1, x1]=ode45('human', [0,100], 1000); %数值解
figure(1);
plot(t1,x1,'r');
figure(2);
t=0: 0.1: 100;
x=1000*exp(0.05*t); %精确解
plot(t1,x1,'r',t, x)
```



即可得到数值解的结果.

对比数值解与真实解，两条曲线高度重合.

例2. 两种群竞争模型

假设甲乙两个种群开展竞争，t时刻甲、乙各自的数量分别是 $x(t), y(t)$，则其变化规律可以描述为：

$$\begin{cases} x'(t)=ax\left(1-\dfrac{x}{N}\boxed{-p\dfrac{y}{M}}\right) \\ y'(t)=by\left(1\boxed{-q\dfrac{x}{N}}-\dfrac{y}{M}\right) \end{cases}$$

其中 a、b 表示两个种群的自然增长率；N、M 分别是两物种单独生存时的数量上限；p、q 表示乙对甲和甲对乙的竞争或影响能力.

常微分方程的数值解

首先编写compete.m：

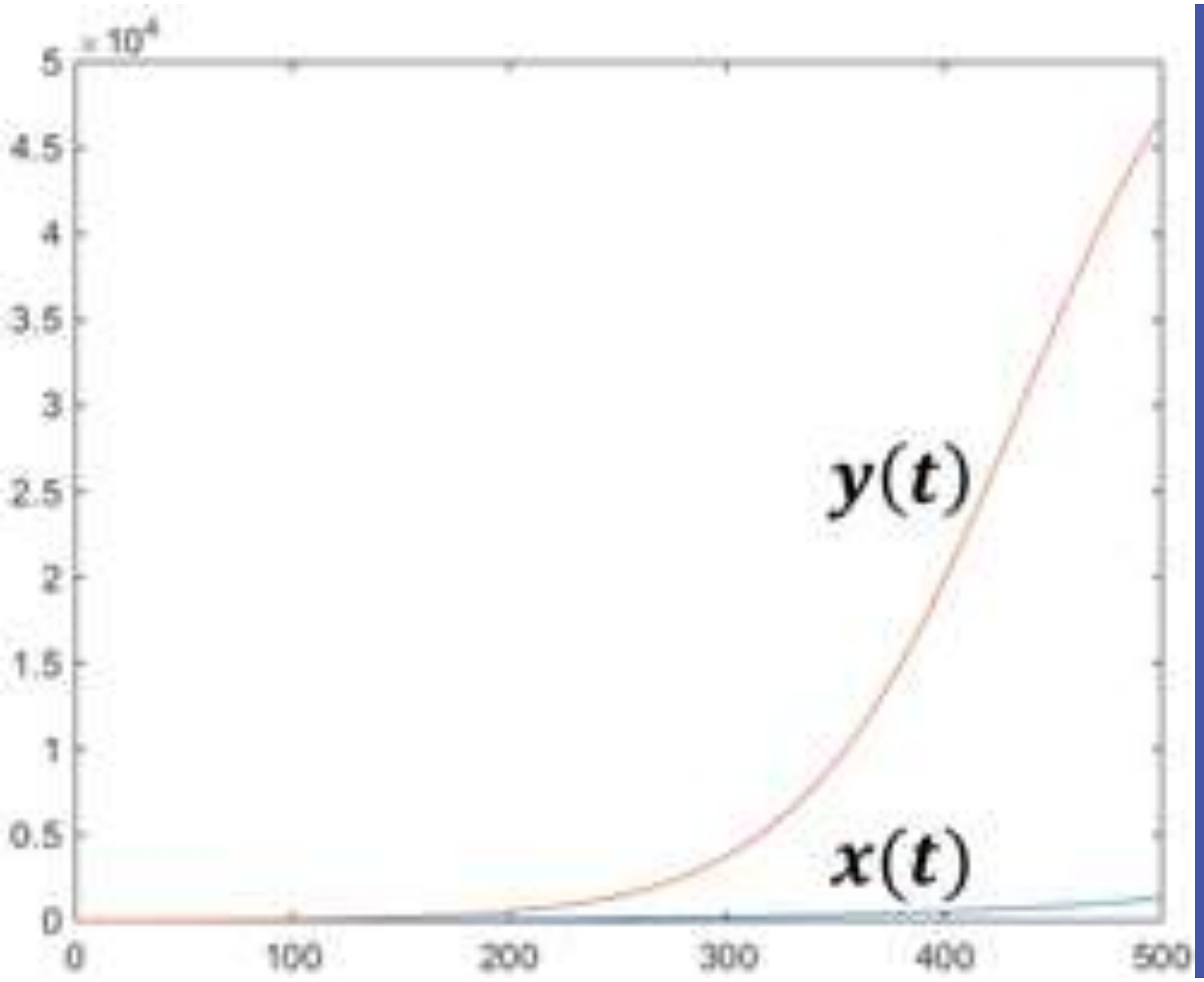
$$\begin{cases} x'(t)=ax(1-\frac{x}{N}-p\frac{y}{M}) \\ y'(t)=by(1-q\frac{x}{N}-\frac{y}{M}) \end{cases}$$

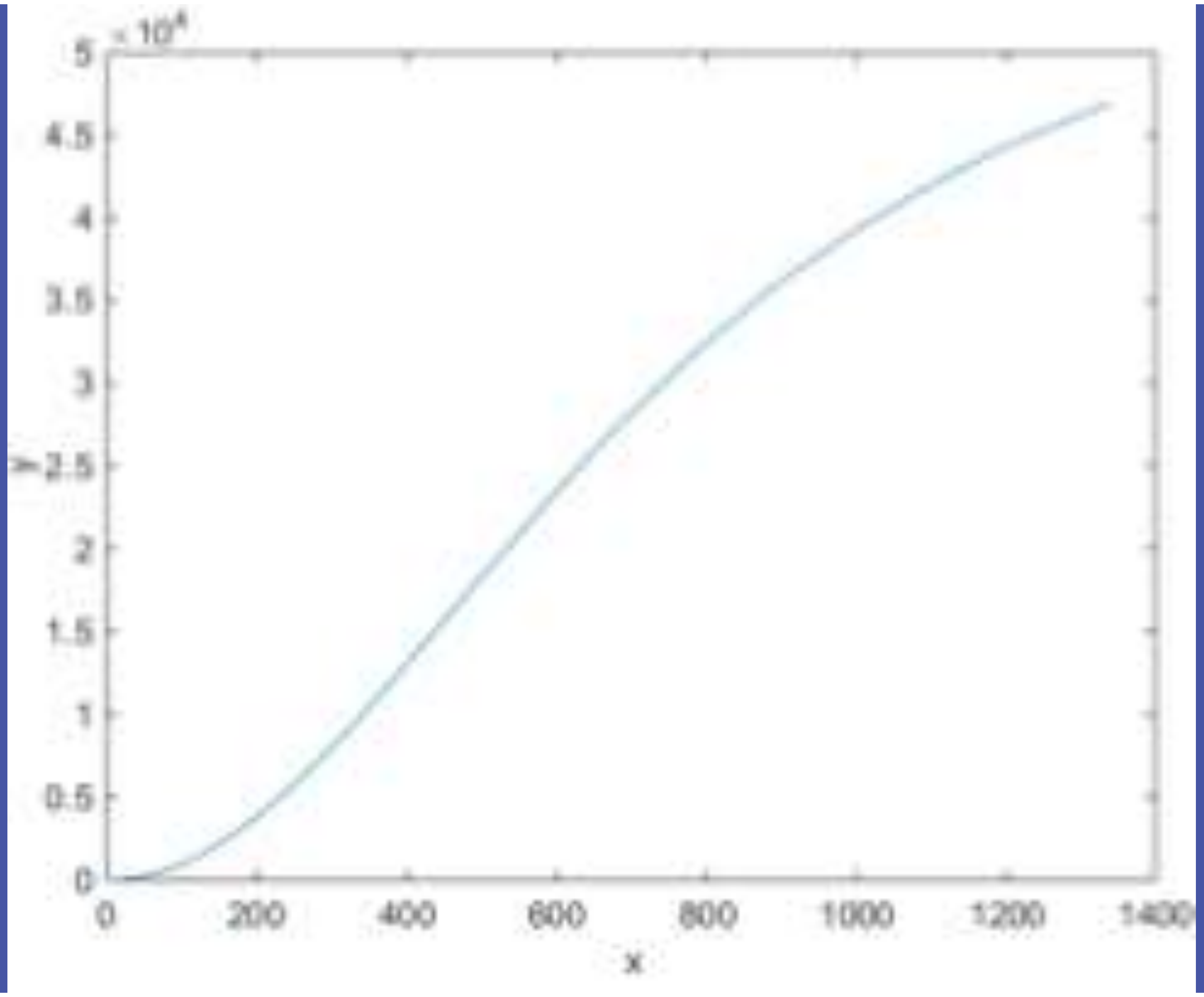
```
function dx=compete(t, x)
dx=zeros(2, 1);
dx(1)=0.01*x(1)*(1-x(1)/50000-0.1*x(2)/60000);
dx(2)=0.02*x(2)*(1-0.2*x(1)/50000-x(2)/60000);
```

注：用x1、x2代表方程组中的x、y. 乙的增长率和上限都占优. 然后输入命令：

```
>> [t, x]=ode45('compete', [0, 500], [10, 10]);
%%[0, 500]是t的取值范围，[10, 10]是初值
plot(t, x(:, 1), t, x(:, 2))          %%绘制t-x和t-y曲线.
figure(2); plot(x(:, 1), x(:, 2))     %%绘制x-y曲线图.
```

常微分方程的数值解





对比可以看出，在乙增长率和上限都占优的情况下，甲的数量增长会受到压制.

常微分方程的数值解

对于常微分方程模型而言，求解往往不是最终目的，更重要的是模型中的参数变化，会对解的性态产生怎样的影响.

因此在求数值解之前，最好能利用常微分方程定性理论对参数的影响做一界定，这样才能更加全面地了解事物内在的运行规律.

练　习

1. 求解 $\dfrac{dy}{dx}+2xy=xe^{-x^2}$，并验证其正确性.

2. 求$x\dfrac{dy}{dx}+y-e^x=0$, $y(1)=2e$的初值解并画出解的图像.

3. 求解 $\begin{cases} x'(t)=2x+y \\ y'(t)=3x+4y \end{cases}$.

4. 用讲过的几种命令分别求$x'(t)=x+2x^2+3x^3$, $x(0)=1$在[0, 10]内的数值解，并分别绘图、分析其区别。

常微分方程的数值解

```
function P=EulerE4(x0,y0,b,h)
%Euler法求解常微分方程
%x0,y0--初值，b-x0取值的右端点，h—步长
n=(b-x0)/h; n=round(n);  %取整
X=zeros(n,1);Y=zeros(n,1);
k=1;
X(k)=x0;Y(k)=y0;
for k=1:n
    X(k+1)=X(k)+h;
Y(k+1)=Y(k)+h*(Y(k)+2*Y(k)^2+3*Y(k)^3);%方程
    k=k+1;
end
plot(X,Y,'mp')
Xlabel('自变量X'), ylabel('因变量y');
```

```
>> x0=0;
y0=1;
b=0.1;
h=0.005;
P=EulerE4(x0,y0,b,h)
```