大学数学实验

实验4 常微分方程数值解

(Numerial Solutions for ODE -- Ordinary Differential Equations)

清 华 大 学 数 学 科 学 系

为什么要学习微分方程数值解

- 微分方程是研究函数变化规律的重要工具，有着广泛的应用。如：

物体的运动，电路的电压，人口增长的预测

- 许多微分方程没有解析解，数值解法是求解的重要手段，如

$$\frac{dy}{dx} = y^2 + x, \qquad \begin{aligned}\dot{x}(t) &= -axy \\ \dot{y}(t) &= axy - by\end{aligned}$$

实验4的基本内容

1. 两个最常用的数值算法:
   - 欧拉（Euler）方法
   - 龙格-库塔（Runge-Kutta）方法
2. 龙格-库塔方法的MATLAB实现
3. 实际问题用微分方程建模，并求解

*4. 数值算法的收敛性、稳定性与刚性方程

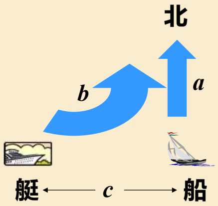
实验4的基本内容

1. 数值微分，及其MATLAB实现
2. ODE问题提法；最常用的数值算法:
   - 欧拉（Euler）方法
   - 龙格-库塔（RK：Runge-Kutta）方法
3. 龙格-库塔方法的MATLAB实现
4. 数值算法的收敛性、稳定性与刚性方程
5. 实际问题用微分方程建模，并求解

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。

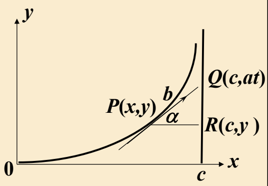


实例1 海上缉私

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

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



由方程无法得到*x*(*t*), *y*(*t*)的解析解
需要用数值解法求解

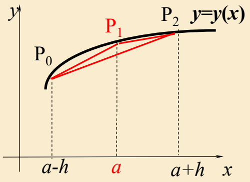
数值微分：数值导数

用离散方法(差商)近似计算函数$y=f(x)$在点$x=a$的导数值

$$f'(a)\cong\frac{f(a+h)-f(a)}{h}$$ 前差公式

$$f'(a)\cong\frac{f(a)-f(a-h)}{h}$$ 后差公式

$$f'(a)\cong\frac{f(a+h)-f(a-h)}{2h}$$ 中点公式



哪个计算公式更好？---- 判断标准？分析方法？

数值导数：误差分析

泰勒展开：$f(a\pm h)=f(a)\pm hf'(a)+\frac{h^2}{2}f''(a)\pm O(h^3)$

$$\Rightarrow\ f'(a)-\frac{f(a+h)-f(a)}{h}=-\frac{h}{2}f''(a)-O(h^2)$$

$f'(a)\cong\frac{f(a+h)-f(a)}{h}$ 前差公式 误差为$O(h)$

$f'(a)\cong\frac{f(a)-f(a-h)}{h}$ 后差公式 误差为$O(h)$

$f'(a)\cong\frac{f(a+h)-f(a-h)}{2h}$ 中点公式 误差为$O(h^2)$

投票 最多可选1项

设置

在区间$[x_0, x_n]$上采用等距节点的中点公式计算数值导数，但端点x_0, x_n可能无法采用中点公式（如区间外的函数定义未知），你觉得应该怎么处理比较好?

- A 左端点改用前差公式、右端点改用后差公式
- B 左端点改用后差公式、右端点改用前差公式
- C 左右端点分别采用其他误差为$O(h^2)$的公式
- D 无所谓，见机行事就行

提交

三点公式

函数$y=f(x)$在等间距h的分点$x_0<x_1<\dots<x_n$ $\quad x_k=x_0+kh$
上用离散数值表示为$y_0, y_1, \dots, y_n$

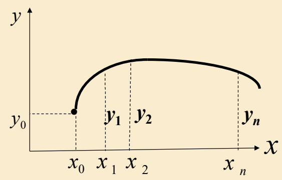
在中间点$x_1, \dots, x_{n-1}$

$$f'(x_k)\cong\frac{y_{k+1}-y_{k-1}}{2h},\quad k=1,2,\cdots n-1$$

在两端点x_0, x_n

$$f'(x_0)\cong\frac{-3y_0+4y_1-y_2}{2h},$$

$$f'(x_n)\cong\frac{y_{n-2}-4y_{n-1}+3y_n}{2h}$$



三点公式，误差为$O(h^2)$

MATLAB 实现

差分

Y = diff(X)	计算差分，步长为 1 的(右)导数
Y = diff(X,n,dim)	沿X的dim维度计算 n 次差分

梯度（导数）

FX = gradient(F, h) 计算向量F的导数(缺省步长h=1)
中点公式（端点前差或后差）

[FX,FY] = gradient(F) 计算矩阵 F (二元函数)的梯度
(更高维度也类似, 参见help)

符号工具箱：diff 也是求符号导数的命令（略）

MATLAB 实现：多项式

多项式求导（polynomial derivative）

q = polyder(p)	p，q 都是多项式系数(降幂)
q = polyder(a，b)	a(x) * b(x) 的导数
[q,d] = polyder(a，b)	a(x) / b(x) 的导数: q(x) / d(x)

多项式积分（polynomial integration）

q = polyint(p,k)	k 为积分常数 (缺省为0)

定义、 求根、 求值、 乘法、 除法、 部分分式展开

poly; roots; polyval; conv; deconv; residue

“常微分方程初值问题数值解”的提法

设 $y'=f(x,y), y(x_0)=y_0$ 的解 y=y(x) 存在且唯一

不求解析解 $y=y(x)$ （无解析解或求解困难）

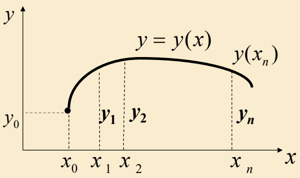
而在一系列离散点 $x_1<x_2<\cdots<x_n<\cdots$

求 $y(x_n)$ 的近似值 $y_n\ (n=1,2,\cdots)$

精确值

常取等距节点，步长 h

$x_n = x_0 + nh$



欧拉方法 $y'=f(x,y), y(x_0)=y_0$

基本思路　在小区间 $[x_n,x_{n+1}]$ 上 $y'=[y(x_{n+1})-y(x_n)]/h$,

$f(x,y)$ 中的 x 取 $[x_n,x_{n+1}]$ 内的某一点

x取不同点 → 各种欧拉公式

$y(x_{n+1})=y(x_n)+hf(x,y(x)), x\in[x_n,x_{n+1}]$

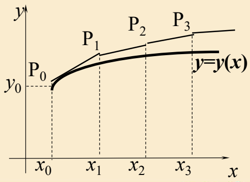
x取左端点 x_n

$y(x_{n+1})=y(x_n)+hf(x_n,y(x_n))$

$y_n\approx y(x_n), y_{n+1}\approx y(x_{n+1})$

$y_{n+1}=y_n+hf(x_n,y_n), n=0,1,\cdots$

向前欧拉公式　　显式公式

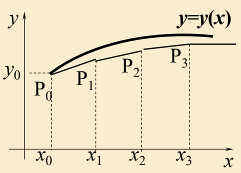


欧拉方法 $y(x_{n+1})=y(x_n)+hf(x,y(x)), x\in[x_n,x_{n+1}]$

x取右端点，$y_n\approx y(x_n), y_{n+1}\approx y(x_{n+1})$

$y_{n+1}=y_n+hf(x_{n+1},y_{n+1}), n=0,1,\cdots$

向后欧拉公式　隐式公式



右端 y_{n+1} 未知，需迭代求解

初值 $y^{(0)}_{n+1}=y_n+hf(x_n,y_n)$

$y^{(k+1)}_{n+1}=y_n+hf(x_{n+1},y^{(k)}_{n+1})$

$k=0,1,2,\cdots, n=0,1,2,\cdots$

$$\lim_{k\to\infty} y^{(k)}_{n+1}=y_{n+1}$$

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

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

校正

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

假定公式右端 …, $y_{n-1}=y(x_{n-1}), y_n=y(x_n), y_{n+1}=y(x_{n+1})$, …

估计算法计算的(左端) y_{n+1} 的误差: $y(x_{n+1})-y_{n+1}$

局部截断误差 (local truncated error)

误差分析　估计欧拉公式的局部截断误差

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

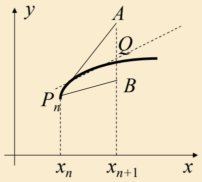
$$y'(x_{n+1}) = y'(x_n+h)$$

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

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

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



误差分析

算法精度的阶（order）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

算法的收敛性

收敛性 步长$h\to 0$时数值解y_n无限接近解析解$y(x_n)$

对于(显式)单步法：只用y_n计算y_{n+1}

单步法 $\begin{cases} y_{n+1}=y_n+h\varphi(x_n,y_n,h) \\ y_0=y(x_0) \end{cases}$

单步法有p阶精度(局部截断误差$O(h^{p+1})$)

$|\varphi(x,y,h)-\varphi(x,\bar{y},h)|\le L|y-\bar{y}|\ (L>0)$ ⇨ **单步法收敛**

⇦ 整体误差 $e_n=y(x_n)-y_n=O(h^p)\to 0\ (h\to 0)$

	向前欧拉公式	改进欧拉公式	4阶龙格-库塔公式
整体误差	$e_n=O(h)$	$e_n=O(h^2)$	$e_n=O(h^4)$

多选题 2分 设置

以下说法正确的是

- A 向前欧拉公式、向后欧拉公式都是显式公式
- B 改进欧拉公式、梯形公式都是隐式公式
- C 改进欧拉公式、梯形公式具有相同的精度阶
- D 多步法既可能是显式公式，也可以是隐式公式
- E 以上都不对

提交

龙格-库塔方法

$$y(x_{n+1})=y(x_n)+hf(x,y(x)),\ x\in[x_n,x_{n+1}]$$

- **向前，向后欧拉公式：**
  用$[x_n, x_{n+1}]$内 1 个点的导数代替 $f(x, y(x))$
- **梯形公式，改进欧拉公式：**
  用$[x_n, x_{n+1}]$内 2 个点导数的平均值代替 $f(x, y(x))$

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代替 $f(x, y(x))$，设法构造出精度更高的计算公式。

龙格-库塔方法的一般形式

$$\begin{cases} y_{n+1}=y_n+h\sum_{i=1}^{L}\lambda_i k_i \\ k_1=f(x_n,y_n) \\ k_2=f(x_n+c_2h,y_n+c_2hk_1) \\ \cdots \\ k_i=f(x_n+c_ih,y_n+c_ih\sum_{j=1}^{i-1}a_{ij}k_j),\quad i=3,4,\cdots,L \end{cases}$$

L级?阶

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

常用的(经典)龙格—库塔公式

$$\begin{cases} y_{n+1}=y_n+\frac{h}{6}(k_1+2k_2+2k_3+k_4) \\ k_1=f(x_n,y_n) \\ k_2=f(x_n+h/2,y_n+hk_1/2) \\ k_3=f(x_n+h/2,y_n+hk_2/2) \\ k_4=f(x_n+h,y_n+hk_3) \end{cases}$$

4级4阶

不足:收敛速度较慢

【介绍】RK方法：级→(最高)阶的关系

Stages	2	3	4	$5 \leqslant m \leqslant 7$	$8 \leqslant m \leqslant 9$	$10 \leqslant m$
Best global error	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{m-1})$	$O(h^{m-2})$	$O(h^{m-3})$

显式RK方法：阶→级的关系

For any positive integer z, an explicit RK method exists with order z and m stages, where

（只是存在性）

$$m = \begin{cases} \frac{3z^2-10z+24}{8}, & z \text{ even} \\ \frac{3z^2-4z+9}{8}, & z \text{ odd} \end{cases}$$

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

$$\begin{cases} y' = f(x,y,z) \\ z' = g(x,y,z) \\ y(x_0)=y_0, z(x_0)=z_0 \end{cases} \xrightarrow{\text{向前欧拉公式}} \begin{cases} y_{n+1} = y_n + hf(x_n,y_n,z_n) \\ z_{n+1} = z_n + hg(x_n,y_n,z_n) \\ n = 0,1,2,\cdots \end{cases}$$

高阶方程需要先降阶为一阶微分方程组

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

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]=ode45(@f, ts, x0)　(4,5)**阶龙格-库塔公式**

4阶法提供候选解，5阶法控制误差(自适应步长)

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:h: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.

参数控制

[t,x]=ode45(@f, ts, x0, opt)　opt：控制参数

[t,x,te,ye,ie]=ode45(…)　事件报告

sol =ode45 (…)　输出结构变量

opt = odeset(…,…)　设定参数(参见help)

例：opt = odeset('RelTol',1e-6, 'AbsTol',1e-9);

更多求解器：ode23, ode23s, ode23t, ode23tb, ode113, ode15s, ode15i (隐式方程)

其他命令：odeget, deval, odextend

多选题　2分　设置

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

用Matlab命令 [t,x]=ode45(@f, t0:h:tf, x0, opt) 求解上述ODE，以下哪些说法是对的?

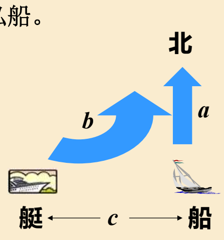
- A **最后计算结果的精度取决于命令中 h 的取值大小**
- B **最后计算结果的精度取决于命令中的参数 opt**
- C **最后计算结果的绝对误差为1e-3，相对误差为1e-6**
- D **输出结果中 t = t0:h:tf**
- E **以上都不对**

提交

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。



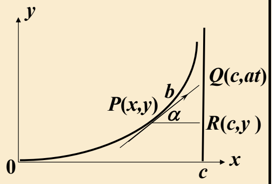
实例1 海上缉私

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

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

由方程无法得到$x(t), y(t)$的解析解
需要用数值解法求解

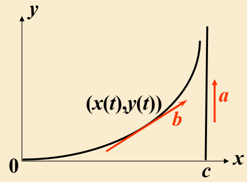


实例1 海上缉私(续)

模型的数值解

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

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



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$y(x)$

```
x0=[0 0];
a=20;b=40;c=15;
[t,x]=ode45(@jisi,ts,x0,[],a,b,c);
%exact solution x1=c
y1=a*t;

%Creat the function for jisi.m
%Let x(1)=x, x(2)=y
function dx=jisi(t,x,a,b,c)
s=sqrt((c-x(1))^2+(a*t-x(2))^2);
dx=[b*(c-x(1))/s;b*(a*t-x(2))/s];
```

```
%output t,x(t),y(t)
[t,x,y1]
%draw x(t),y(t)
plot(t,x),grid,
gtext('x(t)','FontSize',16),
gtext('y(t)','FontSize',16),pause
%draw y(x): the position of tatch js
plot(x(:,1),x(:,2),'r*'),grid
xlabel('x','FontSize',16),
ylabel('y','FontSize',16)
```

MATLAB 6.5.1.lnk
jisi.m, seajisi.m

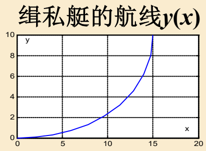
实例1 海上缉私(续)

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)

模型的数值解

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

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

累积误差较大

提高精度!

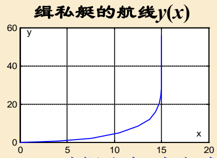
实例1 海上缉私(续)

模型的数值解

```
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？

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1 海上缉私（续） 模型的解析解

$$\frac{dx}{dt}=\frac{b(c-x)}{\sqrt{(c-x)^2+(at-y)^2}},\quad \frac{dy}{dt}=\frac{b(at-y)}{\sqrt{(c-x)^2+(at-y)^2}}$$

$$\Rightarrow \boxed{\frac{dy}{dx}=\frac{at-y}{(c-x)}} \Rightarrow (c-x)\frac{dy}{dx}+y=at \Rightarrow \boxed{(c-x)\frac{d^2y}{dx^2}=a\frac{dt}{dx}}$$

$$\frac{ds}{dt}=b,\quad ds=\sqrt{(dx)^2+(dy)^2}=b\,dt \Rightarrow \boxed{(c-x)\frac{d^2y}{dx^2}=\frac{a}{b}\sqrt{1+\left(\frac{dy}{dx}\right)^2}}$$

令 $p=\frac{dy}{dx}$ $\Rightarrow$ $\boxed{(c-x)\frac{dp}{dx}=k\sqrt{1+p^2},\ k=\frac{a}{b}}$, $p(0)=0$

实例1 海上缉私（续） 模型的解析解

$$(c-x)\frac{dp}{dx}=k\sqrt{1+p^2},\quad p(0)=0 \Rightarrow \begin{cases}\sqrt{1+p^2}+p=\left(\frac{c-x}{c}\right)^{-k}\\ \sqrt{1+p^2}-p=\left(\frac{c-x}{c}\right)^{k}\end{cases} \Rightarrow p=\frac{1}{2}\left[\left(\frac{c-x}{c}\right)^{-k}-\left(\frac{c-x}{c}\right)^{k}\right]$$

$k=a/b<1$, $y(0)=0$

$$\Rightarrow y=\frac{c}{2}\left[\frac{1}{1+k}\left(\frac{c-x}{c}\right)^{1+k}-\frac{1}{1-k}\left(\frac{c-x}{c}\right)^{1-k}\right]+\frac{kc}{1-k^2}$$

缉私艇的航线 $y(x)$ 的解析解

$x=c$时 $y=\frac{kc}{1-k^2}=\frac{abc}{b^2-a^2}$ 缉私艇追上走私船的y坐标

缉私艇追上走私船的时间: $t_1=\frac{bc}{b^2-a^2}$

$a=20, b=40, c=15 \rightarrow t_1=0.5$　　$a=35, b=40, c=15 \rightarrow t_1=1.6$

实 例 2 弱 肉 强 食

问题 自然界中同一环境下两个种群之间的生存方式

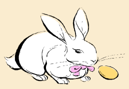
相互竞争　相互依存　弱肉强食

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？



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

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

Volterra模型

$x(t), y(t)$无解析解

实例2 弱肉强食 模型的数值解

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

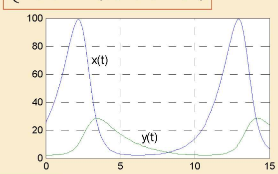
$r=1, d=0.5$

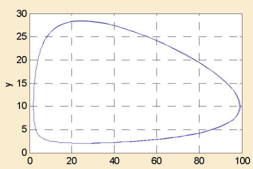
$a=0.1, b=0.02$

$x_0=25, y_0=2$

MATLAB 6.5.1.lnk

shier.m, shier1.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: $\bar{x}\approx 25,\ \bar{y}\approx 10$

实例2 弱肉强食 模型的解析解

$$\begin{cases}\dot{x}=(r-ay)x\\ \dot{y}=-(d-bx)y\end{cases} \Rightarrow \boxed{\frac{dx}{dy}=\frac{(r-ay)x}{-(d-bx)y}} \Rightarrow \frac{-d+bx}{x}dx=\frac{r-ay}{y}dy$$

$\Rightarrow$ 相轨线 $(x^d e^{-xx})(y^r e^{-ay})=c$　c由初始条件确定

可以证明 相轨线是封闭曲线（c在一定范围内） $\Leftrightarrow$ $x(t), y(t)$是周期函数（周期记作T）

求$x(t),y(t)$一周期的平均值: $\bar{x}$, $\bar{y}$

$\dot{y}(t)=-(d-bx)y \Rightarrow x(t)=(\dot{y}/y+d)/b$

$$\Rightarrow \boxed{\bar{x}=\frac{1}{T}\int_0^T x(t)dt}=\frac{1}{T}\left(\frac{\ln y(T)-\ln y(0)}{b}+\frac{dT}{b}\right) \Rightarrow \bar{x}=\frac{d}{b}$$

实例2 弱肉强食 模型的解析解

$x(t),y(t)$一周期的平均值: $\bar{x}=\frac{d}{b}$ $\bar{y}=\frac{r}{a}$

$r=1, d=0.5, a=0.1, b=0.02$ $\bar{x}=25, \bar{y}=10$ 与计算结果同

结果解释

$$\begin{cases}\dot{x}=(r-ay)x\\ \dot{y}=-(d-bx)y\end{cases}$$

r ~食饵增长率

a ~捕食者对食饵的捕获能力

d ~捕食者死亡率

b ~食饵对捕食者的喂养能力

$r\uparrow, a\downarrow \Rightarrow \bar{y}\uparrow$ $d\uparrow, b\downarrow \Rightarrow \bar{x}\uparrow$ 既相互制约 又相互依存

数值算法的稳定性

稳定性 计算中舍入误差不会随步数的增加无限增大

y_n的误差ε_n $|\varepsilon_{n+k}|\le|\varepsilon_n|,\ k=1,2,\cdots$ ⇒ 算法稳定

$y'=f(x,y)$ ⇒ $y'=f(x^*,y^*)+f_x(x^*,y^*)(x-x^*)+f_y(x^*,y^*)(y-y^*)$

$y'=-\lambda y,\ \lambda>0$ (特征根 -λ) ⇒ $y=ce^{-\lambda x}$ λ>0→微分方程稳定

向前欧拉公式 $y_{n+1}=y_n+hf(x_n,y_n)=(1-h\lambda)y_n$ ⇒ $\varepsilon_{n+1}=(1-\lambda h)\varepsilon_n$

$|\varepsilon_{n+k}|\le|\varepsilon_n|$ ⇒ $|1-h\lambda|\le 1$ ⇒ $h\le 2/\lambda$

向后欧拉公式 $y_{n+1}=y_n-h\lambda y_{n+1}$ ⇒ $\varepsilon_{n+1}=\frac{1}{1+h\lambda}\varepsilon_n$ ⇒ h任意

经典龙格-库塔公式 $h\le 2.785/\lambda$

变量变换？

$y'=f(x,y)$ ⇒ $y'=f(x^*,y^*)+f_x(x^*,y^*)(x-x^*)+f_y(x^*,y^*)(y-y^*)$

$y'=-\lambda y,\ \lambda>0$ (特征根 -λ) ⇒ $y=ce^{-\lambda x}$ λ>0→微分方程稳定

不妨记为 y' = A + Bx + Cy (C<0)

做变换 Y = K + Bx + Cy (K待定)

→ Y' = B + Cy'
= B + C(A+Bx+Cy)
= B + CA – CK + C(K+Bx+Cy)
= B + CA – CK + CY

令 K = A + B / C ⇒ Y' = CY

刚性现象与刚性方程

现象 振动系统或电路系统的数学模型

$\ddot{x}+k\dot{x}+rx=f(t)\ (k,r>0), x(0)=a, \dot{x}(0)=b$ $k=2000.5, r=1000, a=1, b=-1999.5, f(t)=1$

⇒ $x(t)=e^{-2000t}-e^{-t/2}+1$ 瞬态解与稳态解

e^{-2000t} ~快瞬态解 计算到$t=0.005$时已衰减到4.5×10^{-5}

$e^{-t/2}$ ~慢瞬态解 计算到$t=20$时才衰减到4.5×10^{-5}

求稳态解 精度达到10^{-4}需算到$t=20$(由慢瞬态解$\lambda=1/2$决定)

选取步长h由快瞬态解$\lambda=2000$决定

龙格-库塔公式 $h<2.785/2000=0.0014$ $t=20$需14286步

快、慢瞬态解的特征根相差悬殊 ⇒ 刚性现象(Stiff)

刚性现象与刚性方程

刚性方程 振动、电路及化学反应中的线性常系数方程组

$$\dot{x}(t)=Ax+f(t),\quad x\in R^n$$

A的特征根λ_k ($k=1,2,\ldots,n$) 的实部$Re(\lambda_k)<0$

$|\mathrm{Re}(\lambda_k)|$大 ~快瞬态解

$|\mathrm{Re}(\lambda_k)|$小 ~慢瞬态解

$s=\frac{\max_k|\mathrm{Re}(\lambda_k)|}{\min_k|\mathrm{Re}(\lambda_k)|}$ ~刚性比 $s>10$ ⇒ 刚性方程

快瞬态解→步长充分小

慢瞬态解→积分区间长 ⇒ 传统的数值方法无能为力

刚性非线性方程组 ⇒ 线性常系数方程组

线性化方法

刚性现象与刚性方程

刚性方程的MATLAB求解

ode23, ode45 解刚性方程的困难

步长自动变小 计算时间很长

求解刚性方程的命令: ode23s, ode15s 等 (用法相同)

例
$$\begin{cases}\dot{x}_1=x_1+2x_2\\ \dot{x}_2=-(10^6+1)x_1-(10^6+2)x_2\\ x_1(0)=10^6/4,\ x_2(0)=10^6/4-1/2\end{cases}$$

特征根$\lambda_1=-1, \lambda_2=-10^6$ 刚性比 $s=10^6$

解析解
$$\begin{cases}x_1(t)=(\frac{10^6}{4}+1)e^{-t}-e^{-10^6t}\\ x_2(t)=-(\frac{10^6}{4}+1)e^{-t}+\frac{(10^6+1)}{2}e^{-10^6t}\end{cases}$$

MATLAB 6.5.1.lnk

Stiff.m, stiff1.m

大学数学实验

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；