Chapter 2

常微分方程初值问题的数值求解

西安交通大学 理学院
蒋臣威

本章内容

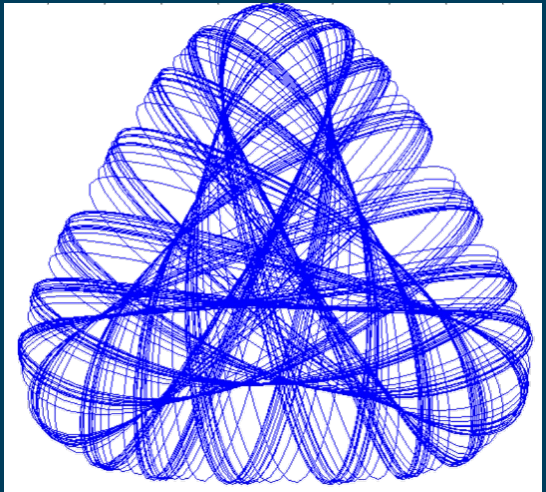
1 简单方法

2 多步法和隐式法

3 Runge-Kutta方法

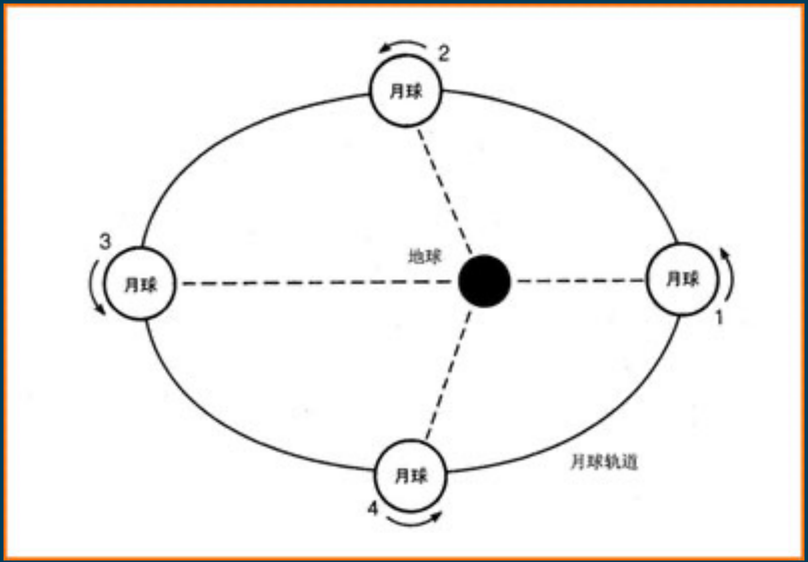
4 稳定性问题

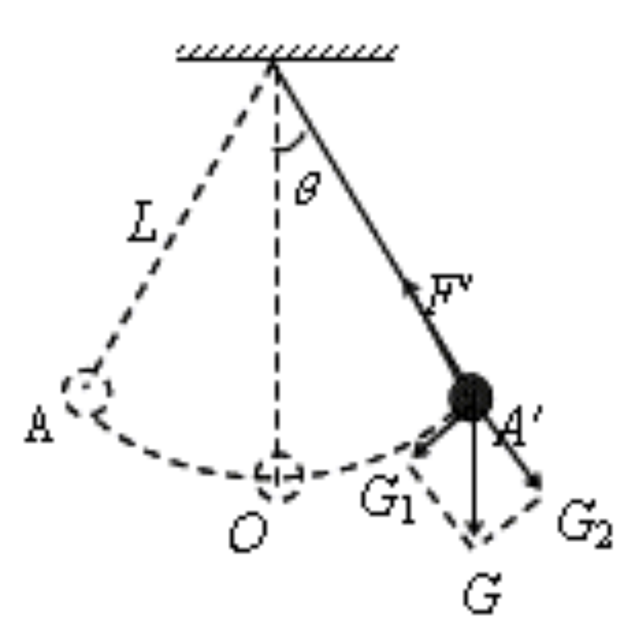
5 动力学中的有序和混沌



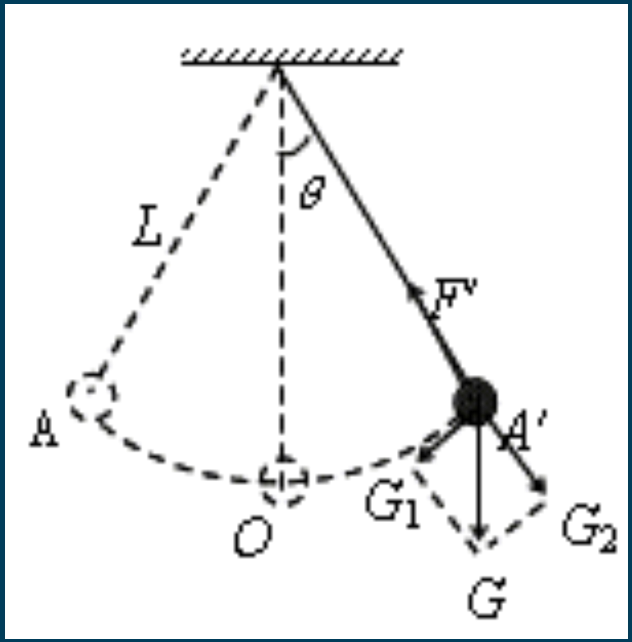
常微分方程的数值解

- 自然界中很多问题的描述，在数学中往往都归结为常微分方程的求解问题。
- 例如天文学中研究星体运动、空间技术中研究物体飞行、物理学中研究单摆的运动等，都需要求解常微分方程。





虽然求解微分方程有许多解析方法,但解析方法通常只能够求解一些**特殊类型的方程**或**微分方程的一些特殊的情况**。从实际意义上来讲我们更关心的是某些特定的自变量在某一个定义范围内的一系列离散点上的近似值。



单摆运动可以用如下常微分方程描述

$$mL\ddot{\theta} = -mg\sin\theta$$

当 $\theta_0 \le 5^\circ$ 时，$\sin\theta \approx \theta$

其解析解为

$$\theta(t) = \theta_0 \cos\omega t,\ \omega = \sqrt{g/L}$$

当 θ_0 很大时，上述常微分方程没有**解析解**，必须借助**数值方法**求解该方程，以描述单摆的运动。

一般地，*n* 阶常微分方程，通过引入若干**辅助函数**可以写成 *n* 个耦合的一阶常微分方程。

以物理学中常见的一个常微分方程（用于描述一维运动粒子）为例

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}=f(x,t) \quad\Rightarrow\quad \begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t}=\dfrac{p}{m}\\[2ex] \dfrac{\mathrm{d}p}{\mathrm{d}t}=f(x,t)\end{cases}$$

一组 M 个耦合的一阶常微分方程可表示为

$$\frac{\mathrm{d}\vec{y}}{\mathrm{d}x}=\vec{f}(x,\vec{y})$$

x 是自变量

$\vec{y}$ 是一组 M 个因变量

因此，只要详细地讨论一阶常微分方程的数值方法就可以了!

$$\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$$

常微分方程的分类

初值问题

给定待求函数在某个初始点上的值

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = f(x,y) \\ y\big|_{x=a} = y_0 \end{cases}$$

边界值问题

在自变量的两个端点上对待求函数施加约束，如函数值约束、导数值约束...

$$\begin{cases} y'' + \dfrac{\pi^2}{4} y + \dfrac{\pi^2}{4} = 0 \\ y(0) = 0,\ y(1) = 1 \end{cases}$$

本征值问题

一类特殊的含有参数的边值问题。只有在参数取特定值时，方程才有非零解

$$\begin{cases} \dfrac{\mathrm{d}^2\varphi(x)}{\mathrm{d}x^2} + k^2\varphi(x) = 0 \\ \varphi(0) = \varphi(1) = 0 \end{cases}$$

初值问题的提法

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = f(x,y) \\ y\big|_{x=a} = y_0 \end{cases} \Rightarrow \; y = y(x),\;\; x \in [a,b]$$

数值问题的提法：求节点x_k处的近似值 y_k ，步长$h_k = x_{k+1} - x_k$，为了方便，常取h不变，这样 $N=(b-a)/h$ 。

策略

1) **离散化：将区间 [a, b] 分为 N 个等间隔的子区间，每个区间宽度为 $h=(b-a)/N$**

2) 寻找一个递推关系: 把 y_n 同 $\{y_{n-1}, y_{n-2}, \ldots\}$ 联系起来。

一、简单方法

$$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+\frac{h^3}{6}f'''(x_0)+O(h^4)$$

1. Euler法

向前差分(商) $$f'(x_0)=\frac{f(x_0+h)-f(x_0)}{h}+O(h)$$

$$\Rightarrow\quad f(x_0+h)=f(x_0)+hf'(x_0)+O(h^2)$$

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)\\ y\big|_{x=a}=y_0\end{cases}$$

$$\Rightarrow\quad \begin{cases}y_{n+1}=y_n+hf(x_n,y_n)+O(h^2)\\ y(x_0)=y_0 \qquad n=1,2,\cdots,N\end{cases}$$

—— Euler法

局部误差为 $O(h^2)$

精度太低！

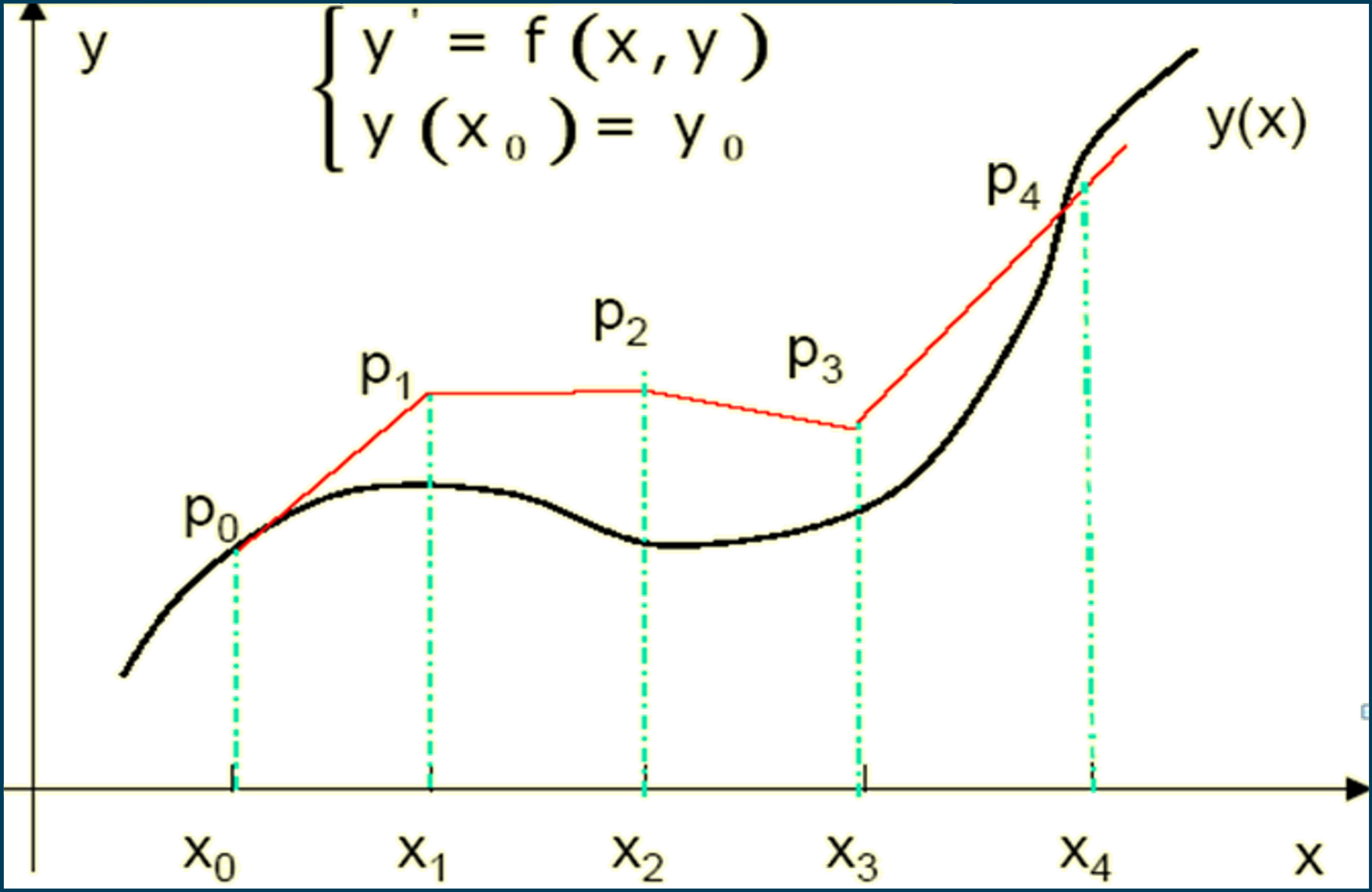
全局误差为 $NO(h^2)\approx O(h)$

通过欧拉方法的讨论

弄清常微方程初值问题数值解法的一些基本概念和构造方法的思路.

欧拉法的几何意义

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) + O(h^2) \\ y(x_0) = y_0 \end{cases}$$



例题1. 考虑下述常微分方程（一阶常微分方程）和初值条件

用 …… 长h，并

与 ……

解 ……

Euler法

h	y(1)	y(3)
0.500	−.143469	.011109
0.200	−.046330	.006519
0.100	−.021625	.003318
0.050	−.010453	.001665
0.020	−.004098	.000666
0.010	−.002035	.000333
0.005	−.001014	.000167
0.002	−.000405	.000067
0.001	−.000203	.000033

见Matlab程序 chapter2_example_1_Euler.m

牛顿动力学方程（二阶常微分方程）

一维运动粒子的方程为 $m\dfrac{\mathrm{d}^2x}{\mathrm{d}t^2}=f(x,t)$

用欧拉法写出具体的递推关系 $y_{n+1}=y_n+hf(x_n,y_n)$

$$\begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t}=\dfrac{p}{m}\\[2ex]\dfrac{\mathrm{d}p}{\mathrm{d}t}=f(x,t)\end{cases}\quad\Longrightarrow\quad\begin{cases}x_{n+1}=x_n+hp_n/m\\p_{n+1}=p_n+hf(x_n,t_n)\end{cases}$$

欧拉法的Matlab实现

标量形式

```
for n = 1:N
  x(n+1) = x(n) + h * p(n) /m
  p(n+1) = p(n) + h * f(x(n), t(n))
end
```

例题2. 一维运动粒子的方程为 $m\dfrac{\mathrm{d}^2x}{\mathrm{d}t^2}=-kx$

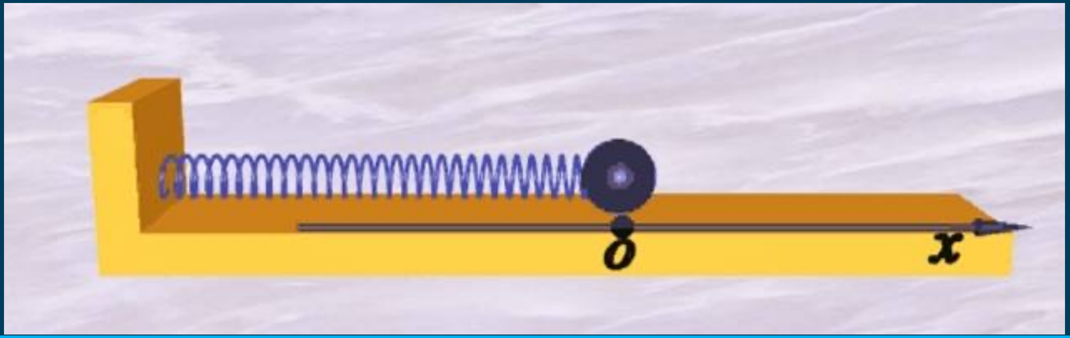
初始条件： $x_0=5;\ p_0=0$

其精确解为： $x=A\cos(w_0t+\phi)=A\cos\left(\sqrt{\dfrac{k}{m}}t+\phi\right)$

$$p=-mAw_0\sin(w_0t+\phi)$$

其中振幅A与初始相位由初始条件决定。

$$\frac{1}{2}kx_0^2+\frac{1}{2}mv_0^2=\frac{1}{2}kA^2 \qquad \phi=\cos^{-1}(x_0/A)$$



见Matlab程序
chapter2_example_2_harmonic_oscillator.m

2. 泰勒级数法

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y)$$

将 y_{n+1} 在 y_n 附近做泰勒展开

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

$$y_n' = f(x_n, y_n) \equiv f_n$$

$$y_n'' = \frac{\mathrm{d}}{\mathrm{d}x} f(x_n, y_n) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f \equiv f_x + f_y f$$

$$\longrightarrow y_{n+1} = y_n + hf_n + \frac{1}{2}h^2[f_x + f_y f]_n + O(h^3)$$

局部误差为 $O(h^3)$　　全局误差为 $NO(h^3) \approx O(h^2)$

这个公式在已知 f 的解析形式时很好用。阶数增大时，变得复杂。

练习：推导三阶泰勒级数法

$$\boxed{y_{n+1} = y_n + ?}$$

$$y_{n+1} = y(x_n + h) = y_n + hy'_n + \frac{1}{2}h^2 y''_n + \frac{1}{6}h^3 y'''_n + O(h^4)$$

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2[f_x + f_y f]_n +$$

$$+\frac{1}{6}h^3[f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y(f_x + f_y f)]_n + O(h^4)$$

例题3. 考虑下述微分方程和初始条件

$$dy/dx = -xy\,;\; y(0) = 1$$

用Euler法与二阶泰勒级数法求解$y(2)$的值，采用不同的步长h，并与精确值比较，计算误差大小。

解答：方程的精确解为

$$y = e^{-x^2/2}$$

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2[f_x + f_y f]_n + O(h^3)$$

$$= y_n - hx_n y_n + \frac{1}{2}h^2[-y_n + x_n^2 y_n] + O(h^3)$$

见Matlab程序

chapter2_example_3_Euler_Taylor.m

通常，建立微分方程的数值算法，有两条途径

(1) 用数值微分的两点公式 $\dfrac{y_{n+1}-y_n}{x_{n+1}-x_n}$ 替代微分方程左边的一阶微商；如Euler法。

(2) 从$x \to x+k$积分恒等式$y'(t)=f(t, y(t))$，得

$$y(x+k)-y(x)=\int_x^{x+k} f(t, y(t))\mathrm{d}t$$

取 $x=x_n$, $k=h$ 有

$$y_{n+1}-y_n=\int_{x_n}^{x_{n+1}} f(t, y(t))\mathrm{d}t$$

下面我们通过第二条途径推导其他数值算法

二. 多步法和隐式法

多步法

对方程 $\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$ 作一步精确积分

$$y_{n+1}=y_n+\int_{x_n}^{x_{n+1}} f(x,y(x))\mathrm{d}x \qquad (1)$$

Lagrange 线性插值：

$$\varphi_1(x)=y_0\frac{x-x_1}{x_0-x_1}+y_1\frac{x-x_0}{x_1-x_0}$$

$$=y_{n-1}\frac{x-x_n}{x_{n-1}-x_n}+y_n\frac{x-x_{n-1}}{x_n-x_{n-1}}=-y_{n-1}\frac{x-x_n}{h}+y_n\frac{x-x_{n-1}}{h}$$

我们用 x_n 和 x_{n-1} 点上的 y 值来提供 f 在所需区间[x_n , x_{n+1}]上的一个线性外插值

$$f=\frac{x-x_{n-1}}{h}f_n-\frac{x-x_n}{h}f_{n-1}+O(h^2) \qquad (2)$$

其中 $f_n=f(x_n,y_n)$ 将（2）带入（1）式，并对x积分可得：

$$\Rightarrow \quad y_{n+1}=y_n+h\left(\frac{3}{2}f_n-\frac{1}{2}f_{n-1}\right)+O(h^3)$$

推导时注意：$x_{n+1}-x_n=x_n-x_{n-1}=h$

—— *Adams-Bashforth二步法*

若以 $f_n, f_{n-1}, f_{n-2}, f_{n-3}$ 的拟合多项式来外插（四次Lagrange插值），得Adams-Bashforth四步法

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + O(h^5)$$

单单关于y_0的值无法启动上述多步法，该方法的启动需要通过其他方法（如Euler法，泰勒级数法等）获得多个f值。

例题4. 考虑下述微分方程和初始条件

$$dy/dx = -xy\,;\; y(0) = 1$$

用Adams-Bashforth二步法求解$y(2)$的值，采用不用的步长h，并与精确值比较，计算误差大小。用Euler法或泰勒级数法产生启动递推关系所需要的y值。

见Matlab程序
chapter2_example_4_Adams_Bashforth.m

隐式法

在区间 $[x_n, x_{n+1}]$ 上过 f_n， f_{n+1} 两点对 f 做线性插值，得

$$f = \frac{x - x_n}{h} f_{n+1} - \frac{x - x_{n+1}}{h} f_n + O(h^2)$$

带入积分公式 $$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y) dx$$

得 $$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] + O(h^3)$$

上式中等号两边都有 y_{n+1} 项，因此称为隐式法。

下面来推导一个典型的隐式法：*Adams-Moulton二步法*

对方程 $\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y)$ 作一步精确积分 $y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y)\mathrm{d}x$

Lagrange 二次插值：

$$\varphi_2(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x)$$

$$l_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$

$$l_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \qquad l_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

我们用 f_{n-1}, f_n 和 f_{n+1} 三点上作二次多项式插值

$$f = \frac{(x-x_n)(x-x_{n-1})}{2h^2} f_{n+1} - \frac{(x-x_{n+1})(x-x_{n-1})}{h^2} f_n + \frac{(x-x_{n+1})(x-x_n)}{2h^2} f_{n-1} + O(h^3)$$

代入积分公式，从 $x_n \to x_{n+1}$ 上对 f 作积分, 得

$$y_{n+1} = y_n + \frac{h}{12}(5 f_{n+1} + 8 f_n - f_{n-1}) + O(h^4)$$

—— *Adams-Moulton二步法*

同样方法用三次多项式插值，可得三步法公式

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) + O(h^5)$$

—— *Adams-Moulton三步法*

Adams-Moulton二步法与三步法都既是多步法又是隐式法

多步法：最初几个格点的启动值需要通过其它的方法如欧拉法、泰勒级数法来获得

隐式法: 不能直接使用，它通常用在预报校正算法中

先通过显式法“预报”一个 y_{n+1}的值，然后利用隐式法来校正得到一个更精确的值。这样的算法有个优点，它可以持续监控积分的精度。

一个常常使用的具有局域误差 $O(h^5)$ 的预报校正算法：

显式的 Adams-Bashforth 四步法 + **Adams-Moulton三步法**

隐式法意味着在每一个积分步都必须解一个方程，会非常耗时。

隐式算法的“预估-校正”思想可以帮助我们推导出著名的**Runge-Kutta**方法。

三、Runge-Kutta方法

问题的提出

(1) 隐式算法的预估-校正，多步法的启动都需要我们寻求一种高阶的单步算法。

(2) 要得到高阶方法的一个直接想法就是用Taylor展开，但这必须计算 y 的高阶微商，而一般情况下，求 $f(x,y)$ 的微商相当麻烦，因此需寻求一种不求高阶微商的方法。

思想：从欧拉法说起

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y)$$

在子区间 $[x_n,\ x_{n+1}]$ 将微分方程写成积分形式

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x$$

平均斜率

中值定理

$$\begin{cases} y_{n+1} = y_n + hK_{ave} \\ K_{ave} = f(x_n + \theta h, y(x_n + \theta h)) \end{cases}$$

欧拉法就是用 x_n 点的斜率近似 $[x_n,\ x_n+1]$区间的平均斜率 K_{ave}

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

改进的欧拉法 – 梯形公式

用 x_n，x_{n+1} 两点斜率K_1和K_2的平均值来近似区间[x_n，x_{n+1}]的平均斜率

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

但是由于 y_{n+1} 待定，因此需要做“预报”，即“预估-校正”

对预估值做校正

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

来自简单Euler法，预估y_{n+1}

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] + O(h^3)$$ 隐式法

上述算法即Euler法与隐式法的结合。

考察改进的欧拉法，可以将其改写为：

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

斜率

一定取K_1 K_2的平均值吗？

步长一定是一个h 吗？

Runge-Kutta法的基本思想

基本思想

通过求不同点上的函数值，作线性组合，构造近似公式，把近似公式和解的Taylor展开式相比较，使其前面若干项相吻合，从而得到相应的高阶方法。

$$\boxed{y_{n+1} = y_n + h\sum_{i=1}^{N} \lambda_i K_i} \qquad (☼)$$

其中
$$\begin{cases} K_1 = f(x_n, y_n) = f_n \\ K_i = f(x_n + hc_i, y_n + hc_i \sum_{j=1}^{i-1} a_{ij} K_j) \end{cases} \quad i = 2,3,\cdots,N$$

参数λ_i, c_i, a_{ij} 选取的原则是要求(☼)的右端在(x_n, y_n)处作Taylor展开后，按 h 的幂次重新整理

$$\tilde{y}_{n+1} = y_n + d_1 h + \frac{1}{2!} d_2 h^2 + \frac{1}{3!} d_3 h^3 + \cdots$$

表示用$y(x_n)$计算得到的近似值

与微分方程的解$y(x)$在x_n点处的Taylor展开式

$$y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!} y''(x_n) + \cdots$$

有尽可能多的项重合。

以计算两个函数值的情况为例

$$y_{n+1} = y_n + h\sum_{i=1}^{N} \lambda_i K_i$$

$$\begin{cases} y_{n+1} = y_n + h\lambda_1 K_1 + h\lambda_2 K_2 \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + hc_2, y_n + hc_2 a_{21} K_1) \end{cases}$$

现在来确定$\lambda_1, \lambda_2, c_2, a_{21}$

$$K_1 = f_n = y'(x_n)$$

二元函数的泰勒展开：

$$f(x_0+h, y_0+k) = f(x_0, y_0) + \left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right) f(x_0, y_0) + \frac{1}{2!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^2 f(x_0, y_0)$$
$$+ \cdots + \frac{1}{n!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^n f(x_0, y_0) + \frac{1}{(n+1)!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^{n+1} f(x_0 + \theta h, y_0 + \theta_k)$$

$$\Rightarrow \quad K_2 = f_n + hc_2(f_x + a_{21} f f_y) + O(h^2)$$

$$\tilde{y}_{n+1} = y_n + h\lambda_1 K_1 + h\lambda_2 K_2$$
$$= y_n + h\lambda_1 f_n + h\lambda_2 (f_n + hc_2(f_x + a_{21} f f_y)) + O(h^3)$$
$$= y_n + h(\lambda_1 + \lambda_2) f_n + h^2 \lambda_2 c_2 (f_x + a_{21} f f_y) + O(h^3)$$

$$y''(x_n) = f'\big|_{(x_n, y_n)} = f_x + f_y f$$

$y(x_{n+1})$在x_n处作Taylor展开

$$y_{n+1} = y(x_n + h) = y_n + hf_n + \frac{h^2}{2!}(f_x + f_y f) + O(h^3)$$

比较得 $\lambda_1 + \lambda_2 = 1, \quad a_{21} = 1, \quad \lambda_2 c_2 = 1/2$

显然，满足上述关系的λ_1, λ_2, c_2, a_{21} 有无穷多组。

选取 $\lambda_1 = \lambda_2 = 1/2, \quad a_{21} = 1, \quad c_2 = 1$

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}K_1 + \frac{h}{2}K_2 + O(h^3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1) \end{cases}$$

Euler法的梯形公式

二阶Runge-Kutta法

若选取 $\lambda_1 = 0, \quad \lambda_2 = 1, \quad a_{21} = 1, \quad c_2 = 1/2$

$$\begin{cases} y_{n+1} = y_n + hK_2 + O(h^3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1) \end{cases}$$

二阶Runge-Kutta法与二阶泰勒级数法比较

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}K_1 + \frac{h}{2}K_2 + O(h^3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1) \end{cases}$$

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2[f_x + f_y f]_n + O(h^3)$$

可以看出，相比二阶泰勒级数法而言，二阶Runge-Kutta法适用性更广，使用也更为方便。

三阶Runge-Kutta法

$$
\begin{cases}
y_{n+1} = y_n + h\lambda_1 K_1 + h\lambda_2 K_2 + h\lambda_3 K_3 \\
K_1 = f(x_n, y_n) \\
K_2 = f(x_n + hc_2, y_n + hc_2 a_{21} K_1) \\
K_3 = f(x_n + hc_3, y_n + hc_3 a_{31} K_1 + hc_3 a_{32} K_2)
\end{cases}
$$

二元函数的泰勒展开：

$$
f(x_0+h, y_0+k) = f(x_0, y_0) + \left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right) f(x_0, y_0) + \frac{1}{2!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^2 f(x_0, y_0)
$$

$$
+ \cdots + \frac{1}{n!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^n f(x_0, y_0) + \frac{1}{(n+1)!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^{n+1} f(x_0 + \theta h, y_0 + \theta_k)
$$

$$
K_2 = f_n + hc_2(f_x + a_{21} f f_y) + \frac{1}{2} h^2 c_2^2 (f_{xx} + a_{21}^2 f_n^2 f_{yy}
$$

$$
+ 2a_{21} f_n f_{xy}) + O(h^3)
$$

$$K_3 = f_n + hc_3(f_x + a_{31}ff_y + a_{32}K_2 f_y) + \frac{1}{2}h^2c_3^2[f_{xx} +$$
$$(a_{31}f + a_{32}K_2)^2 f_{yy} + 2(a_{31}f + a_{32}K_2)f_{xy}] + O(h^3)$$

将上式中的K_2替换掉，注意略去导致最终展开式中的$O(h^4)$项

$$y_{n+1} = y_n + h\lambda_1 K_1 + h\lambda_2 K_2 + h\lambda_3 K_3$$
$$= y_n + h\lambda_1 f_n + h\lambda_2[f_n + hc_2(f_x + a_{21}ff_y) + \frac{1}{2}h^2c_2^2(f_{xx} + a_{21}^2 f_n^2 f_{yy}$$
$$+2a_{21}f_n f_{xy})] + h\lambda_3\{f_n + hc_3(f_x + a_{31}ff_y + a_{32}K_2 f_y) + \frac{1}{2}h^2c_3^2[f_{xx} +$$
$$(a_{31}f + a_{32}K_2)^2 f_{yy} + 2(a_{31}f + a_{32}K_2)f_{xy}]\} + O(h^4)$$

$y(x_{n+1})$在x_n处作Taylor展开

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2[f_x + f_y f]_n +$$
$$+\frac{1}{6}h^3[f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y(f_x + f_y f)]_n + O(h^4)$$

$$\begin{cases} \lambda_1+\lambda_2+\lambda_3=1; \lambda_2 c_2+\lambda_3 c_3=1/2; \\ \lambda_2 c_2 a_{21}+\lambda_3 c_3 a_{31}+\lambda_3 c_3 a_{32}=1/2; \ldots\ldots \end{cases}$$

选取

$$\lambda_1=1/6; \lambda_2=4/6; \lambda_3=1/6;$$

$$c_2=1/2; c_3=1$$

$$a_{21}=1; a_{31}=-1; a_{32}=2$$

可得三阶Runge-Kutta法

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

经典四阶R-K方法（略去繁琐的推导）

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) + O(h^5)$$

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

通常认为四阶R-K法在效率和精度间达到了最好的平衡！

注：

☞ 龙格-库塔法的主要运算在于计算 K_i 的值，即计算 f 的值。Butcher 于1965年给出了计算量与可达到的最高精度阶数的关系：

每步须算K_i的个数	2	3	4	5	6	7	$n\ge 8$
可达到的最高精度	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^4)$	$O(h^5)$	$O(h^6)$	$O(h^{n-2})$

☞ 由于龙格-库塔法的导出基于泰勒展开，故精度主要受解函数的光滑性影响。对于光滑性不太好的解，最好采用低阶算法而将步长h 取小。

例题5： 写出用4阶经典R-K法求初值问题

$$\begin{cases} y' = 8 - 3y \\ y_0 = y(0) = 2 \end{cases}$$

的计算公式。并取步长 $h=0.2$，计算$y\ (0.4)$的近似值

先获得递推关系式 $f(x, y) = 8 - 3y$

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

$$K_1 = f(x_n, y_n) = 8 - 3y_n$$

$$K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1) = 8 - 3(y_n + \frac{h}{2}K_1) = 5.6 - 2.1y_n$$

$$K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2) = 8 - 3(y_n + \frac{h}{2}K_2) = 6.32 - 2.37y_n$$

$$K_4 = f(x_n + h, y_n + hK_3) = 8 - 3(y_n + hK_3) = 4.208 - 1.578y_n$$

$$\boxed{y_{n+1} = 1.2016 + 0.5494y_n}$$

$$\boxed{y_{n+1} = 1.2016 + 0.5494\, y_n}$$

$$y_1 = 1.2016 + 0.5494 y_0 = 1.2016 + 0.5494 \times 2 = 2.3004$$

$$y_2 = 1.2016 + 0.5494 y_1 = 1.2016 + 0.5494 \times 2.3004 = 2.4654$$

逐点求解（即程序计算采用的方法） $f(x, y) = 8 - 3y$

$$K_{11} = f(x_0, y_0) = 8 - 3y_0 = 2$$

$$K_{21} = f(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} K_{11}) = 8 - 3(y_0 + \frac{h}{2} K_{11}) = 8 - 3(2 + \frac{0.2}{2} \times 2) = 1.4$$

$$K_{31} = f(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} K_{21}) = 8 - 3(y_0 + \frac{h}{2} K_{21}) = 8 - 3(2 + \frac{0.2}{2} \times 1.4) = 1.58$$

$$K_{41} = f(x_0 + h, y_0 + hK_{31}) = 8 - 3(y_0 + hK_{31}) = 8 - 3(2 + 0.2 \times 1.58) = 1.052$$

$$y_1 = y_0 + \frac{h}{6}(K_{11} + 2K_{21} + 2K_{31} + K_{41})$$

$$= 2 + \frac{0.2}{6} \times (2 + 2 \times 1.4 + 2 \times 1.58 + 1.052) = 2.3004$$

$$f(x,y)=8-3y$$

$$K_{12}=f(x_1,y_1)=8-3y_1=8-3\times 2.3004=1.0988$$

$$K_{22}=f(x_2+\frac{h}{2},y_2+\frac{h}{2}K_{12})=8-3\times(y_1+\frac{h}{2}K_{12})$$

$$=8-3\times(2.3004+\frac{0.2}{2}\times 1.0988)=0.76916$$

$$K_{32}=f(x_1+\frac{h}{2},y_1+\frac{h}{2}K_{22})=8-3(y_1+\frac{h}{2}K_{22})$$

$$=8-3\times(2.3004+\frac{0.2}{2}\times 0.76916)=0.868052$$

$$K_{42}=f(x_1+h,y_1+hK_{32})=8-3(y_1+hK_{32})$$

$$=8-3\times(2.3004+0.2\times 0.868052)=0.5779688$$

$$y_2=y_1+\frac{h}{6}(K_{12}+2K_{22}+2K_{32}+K_{42})=2.4654$$

精确解 $y = \frac{8}{3} - \frac{2}{3}e^{-3x}$

$$y(0.2) = 2.300792243$$

$$y(0.4) = 2.465870525$$

H=0.200000
Y1=2.300400,Y2=2.465440

H=0.100000
Y1=2.172775,Y2=2.300773,
Y3=2.395599,Y4=2.465850

H=0.050000
Y1=2.092861,Y2=2.172787,
Y3=2.241580,Y4=2.300791,
Y5=2.351755,Y6=2.395619,
Y7=2.433374,Y8=2.465869

H=0.010000
Y1=2.019703,Y2=2.038824,
Y3=2.057379,Y4=2.075386
Y5=2.092861,Y6=2.109820,
Y7=2.126277,Y8=2.142248

见Matlab程序
chapter2_example_5_RK.m

常微分方程初值问题数值解法小结

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y)$$
$$\mathrm{y}\big|_{x_0} = y_0$$

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) + O(h^2) \\ y(x_0) = y_0 \end{cases}$$
—— Euler法

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2) + O(h^3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1)) \end{cases}$$
—— 改进的Euler法-梯形公式（二阶Runge-Kutta算法）

$$\begin{cases} y_{n+1} = y_n + hf_n + \frac{1}{2}h^2[f_x + f_y f]_n + O(h^3) \\ y_n' = f(x_n, y_n) \equiv f_n \\ y_n'' = \frac{\mathrm{d}}{\mathrm{d}x} f(x_n, y_n) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f \end{cases}$$
—— 二阶泰勒级数法

常微分方程初值问题数值解法小结

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) + O(h^5) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1) \\ K_3 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_2) \\ K_4 = f(x_n + h, y_n + hK_3) \end{cases}$$

—— ***四阶经典Runge-Kutta法***

例6 用Euler法、二阶泰勒级数法，Euler法的梯形公式以及经典四阶Runge-Kutta法分别求解下述常微分方程

$$\begin{cases} y' = y - \dfrac{2x}{y} & 0 \le x \le 1 \\ y(0) = 1 \end{cases}$$

的数值解，取h=0.1，并与精确值比较。方程的解析解为:

$$y = \sqrt{1 + 2x}$$

见Matlab程序
chapter2_example_6_Euler_Taylor_Tri_RK.m

高阶常微分方程总可以通过引入辅助变量，转化成一阶常微分方程组，以两个方程组成的方程组为例

$$\begin{cases} \dfrac{dy}{dx} = f(x,y,z) \\ \dfrac{dz}{dx} = g(x,y,z) \\ y(a) = y_0 \\ z(a) = z_0 \end{cases} \quad , \quad a \le x \le b$$

Euler公式

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

四阶经典Runge-Kutta公式

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

$$K_1 = \begin{pmatrix} f(x_n, y_n, z_n) \\ g(x_n, y_n, z_n) \end{pmatrix}$$

$K_1^{(1)}$ 表示 K_1 的第一个元素

$$K_2 = \begin{pmatrix} f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1^{(1)}, z_n + \frac{h}{2}K_1^{(2)}) \\ g(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1^{(1)}, z_n + \frac{h}{2}K_1^{(2)}) \end{pmatrix}$$

$$K_3 = \begin{pmatrix} f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2^{(1)}, z_n + \frac{h}{2}K_2^{(2)}) \\ g(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2^{(1)}, z_n + \frac{h}{2}K_2^{(2)}) \end{pmatrix}$$

$$K_4 = \begin{pmatrix} f(x_n + h, y_n + hK_3^{(1)}, z_n + hK_3^{(2)}) \\ g(x_n + h, y_n + hK_3^{(1)}, z_n + hK_3^{(2)}) \end{pmatrix}$$

• 例题7（高阶常微分方程的Runge-Kutta法求解）

习题 2.5 下面两个耦合的一阶方程

$$\frac{\mathrm{d}y}{\mathrm{d}t}=p\,;\ \frac{\mathrm{d}p}{\mathrm{d}t}=-4\pi^2 y \tag{2.26}$$

确定了周期为 1 的简谐运动．通过把上面给出的各个单变量公式之一推广到这个二变量情形，以你选定的任何特定的初始条件积分方程组(2.26)，并且考察系统在 t 取整数值时回到其初始状态的精度．

$$\begin{cases} y_{n+1}=y_n+\dfrac{h}{6}(K_1+2K_2+2K_3+K_4)+O(h^5) \\ K_1=f(x_n,y_n) \\ K_2=f\left(x_n+\dfrac{h}{2},y_n+\dfrac{h}{2}K_1\right) \\ K_3=f\left(x_n+\dfrac{h}{2},y_n+\dfrac{h}{2}K_2\right) \\ K_4=f(x_n+h,y_n+hK_3) \end{cases}$$

chapter2_example_7_erbianliang_RK_test_1.m
chapter2_example_7_erbianliang_RK_test.m
chapter2_example_7_erbianliang_RK_test_2.m
chapter2_example_7_erbianliang_RK.m

比较不同算法程序

例题7-2. 用Euler法和经典四阶R-K算法求解下述常微分方程

$$\begin{cases} y''' - y'' = x \\ y(1) = 8 \\ y'(1) = 7 \\ y''(1) = 4 \end{cases}$$

其解析解为：

$$y = -\frac{x^3}{6} - \frac{x^2}{2} + 2.5x + 6e^{x-1} + \frac{1}{6}$$

对上述方程进行降阶，令 $y_1 = y,\ y_2 = y',\ y_3 = y''$

$$\begin{cases} y_1' = y_2 & y_1(1) = 8 \\ y_2' = y_3 & y_2(1) = 7 \\ y_3' = y_3 + x & y_3(1) = 4 \end{cases}$$

chapter2_example_7_2_sanbianliang_RK.m
chapter2_example_7_2_sanbianliang_RK_test.m

例题7-3.已知两个耦合的马达满足如下方程组：

$$\begin{cases} \dfrac{\mathrm{d}^2 y_1}{\mathrm{d}t^2} + \dfrac{\mathrm{d}y_1}{\mathrm{d}t} + 0.2(\dfrac{\mathrm{d}y_1}{\mathrm{d}t} - \dfrac{\mathrm{d}y_2}{\mathrm{d}t}) = \sin y_1 + \sin t \\ \dfrac{\mathrm{d}^2 y_2}{\mathrm{d}t^2} + \dfrac{\mathrm{d}y_2}{\mathrm{d}t} + 0.2(\dfrac{\mathrm{d}y_2}{\mathrm{d}t} - \dfrac{\mathrm{d}y_1}{\mathrm{d}t}) = \cos y_2 + \sin t \end{cases}$$

初始条件为：t=0时，y1=1；dy1/dt=2；y2=0；dy2/dt=5；

利用经典四阶Runge-Kutta法求解上述方程组，画出位移y1，y2和速率dy1/dt, dy2/dt随时间的演化关系图。

chapter2_example_7_3_four_x_RK.m

§ 4 收敛性与稳定性

➢ 1.收敛性

定义 若某算法对于任意固定的 $x = x_i = x_0 + i\,h$，当 $h \to 0$（同时 $i \to \infty$）时有 $y_i \to y(x_i)$，则称该算法是**收敛**的。

例：就初值问题 $\begin{cases} y' = \lambda y \\ y(0) = y_0 \end{cases}$ 考察欧拉公式的收敛性。

解：该问题的精确解为 $y(x) = y_0 e^{\lambda x}$

欧拉公式为 $y_{i+1} = y_i + h\lambda y_i = (1+\lambda h) y_i \Rightarrow y_i = (1+\lambda h)^i y_0$

对任意固定的 $x = x_i = i\,h$，有

$$y_i = y_0(1+\lambda h)^{x_i/h} = y_0[(1+\lambda h)^{1/\lambda h}]^{\lambda x_i}$$

$$\to y_0 e^{\lambda x_i} = y(x_i)$$ ✓

$$\lim_{h\to 0}(1+\lambda h)^{1/\lambda h} = e$$

2. 稳定性

例8：考察初值问题 $\begin{cases} y'(x) = -30y(x) \\ y(0) = 1 \end{cases}$ **在区间[0, 0.5]上的解。分别用欧拉公式和改进的欧拉公式计算数值解（h=0.1）。**

节点 x_i	欧拉公式	改进欧拉法	精确解 $y = e^{-30x}$
0.0	1.0000	1.0000	1.0000
0.1	-2.0000	2.5000	4.9787×10^{-2}
0.2	4.0000	6.2500	2.4788×10^{-3}
0.3	-8.0000	1.5626×10^{1}	1.2341×10^{-4}
0.4	1.6000×10^{1}	3.9063×10^{1}	6.1442×10^{-6}
0.5	-3.2000×10^{1}	9.7656×10^{1}	3.0590×10^{-7}

What is wrong ??!

见Matlab程序
chapter2_example_8_stable_Euler_tri.m
chapter2_example_8_stable_Euler_tri_RK.m

稳定性的另一个例子

Euler公式从向前差分公式获得，如果从其他差分形式出发以对称差分(中心差分)近似 $f' \approx \dfrac{f_{+1} - f_{-1}}{2h}$

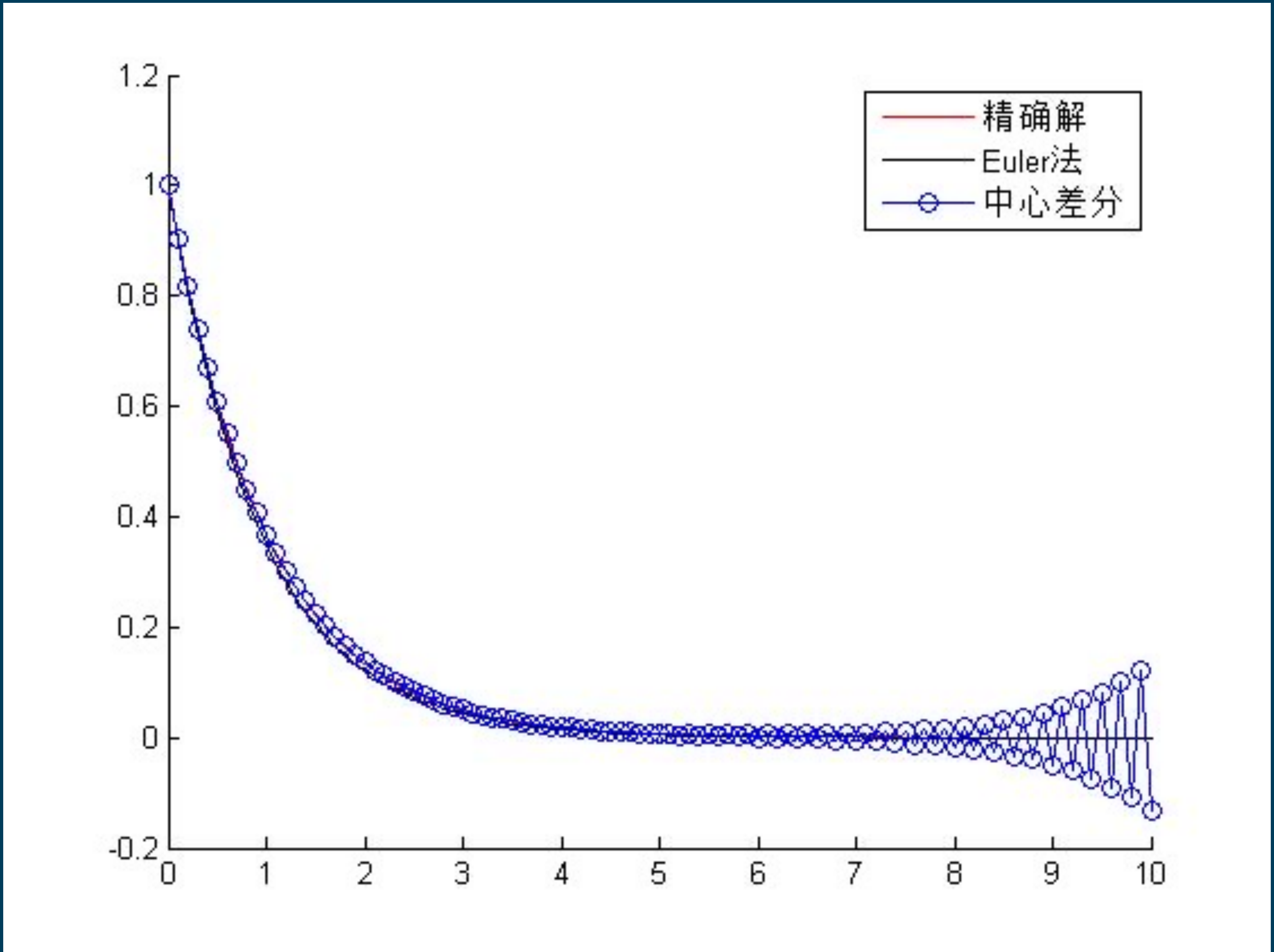
来逼近 $y' = f(x, y)$**中的导数，可得到由三项构成的一个递推关系(二阶方法)**

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) + O(h^3)$$ 多步法

用于例9 $\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = -y \\ y\big|_{x=0} = 1 \end{cases}$ **其精确解为** $y = e^{-x}$

取步长 h**=0.2 or 0.1，** $y_0 = 1,\ \ y_n = ?$ $\qquad y_1 = \begin{cases} e^{-h} \\ 1 - h + \dfrac{1}{2}h^2 + O(h^3) \end{cases}$

分别用Euler法和上述多步法求解



见Matlab程序
chapter2_example_9_stable.m

定义 若某算法在计算过程中任一步产生的误差在以后的计算中都逐步衰减，则称该算法是绝对稳定的。

一般分析时为简单起见，只考虑试验方程

$$y' = \lambda y$$

常数，可以是复数

对于给定步长$h>0$，在计算 y_n 时引入了误差ρ_n ，若这个误差在计算后面的y_{n+i} $(i=1,2,\ldots)$ 中所引起的误差按绝对值均不增加，就说该数值方法对于这个步长 h 和λ 常数是绝对稳定的。为保证方法的绝对稳定，步长 h 和λ 就有其相应的允许范围，称其为该方法的**绝对稳定区域**。

以Euler方法为例 $y_{n+1} = y_n + \lambda h y_n$

误差方程 $\rho_{n+1} = \rho_n(1+\lambda h)$ $\rho_{n+1}/\rho_n = 1+\lambda h$

绝对稳定区域 $|1+\lambda h| \le 1$

对于例8中的存在的误差增大现象，$\lambda = -30$

绝对稳定区域为 $|1-30h| \le 1$ ⇨ $0 \le h \le 1/15$

h=0.1显然不在其绝对稳定区域内，可尝试h=0.05

在例9中，当我们使用中心差分推得的递推公式时，

$$y_{n+1} = y_{n-1} - 2hy_n$$

误差方程为 $\rho_{n+1} = \rho_{n-1} - 2h\rho_n$ $\quad\quad \dfrac{\rho_{n+1}}{\rho_n} - \dfrac{\rho_{n-1}}{\rho_n} = -2h$

令 $K_n = \dfrac{\rho_n}{\rho_{n-1}}$ $\Rightarrow$ $K_{n+1} - \dfrac{1}{K_n} = -2h$

$K_{n+1} \sim K_n$ $\quad\quad K_n^2 + 2hK_n - 1 = 0$

$$K_n = -h \pm \sqrt{1+h^2}$$

$$\left| -h \pm \sqrt{1+h^2} \right| \le 1$$

由此，不论 h 取多少，都不能保证绝对稳定。

推导四阶经典R-K算法的绝对稳定区域

以经典R-K法为例,应用于实验方程 $\dfrac{\mathrm{d}y}{\mathrm{d}x}=\lambda y$ 有

$$\begin{cases} y_{n+1}=y_n+\dfrac{h}{6}(K_1+2K_2+2K_3+K_4)+O(h^5) \\ K_1=f(x_n,y_n) \\ K_2=f(x_n+\dfrac{h}{2},y_n+\dfrac{h}{2}K_1) \\ K_3=f(x_n+\dfrac{h}{2},y_n+\dfrac{h}{2}K_2) \\ K_4=f(x_n+h,y_n+hK_3) \end{cases}$$

$$\begin{cases} k_1=\lambda y_n \\ k_2=(\lambda+\dfrac{1}{2}h\lambda^2)y_n \\ k_3=(\lambda+\dfrac{1}{2}h\lambda^2+\dfrac{1}{4}h^2\lambda^3)y_n \\ k_4=(\lambda+h\lambda^2+\dfrac{1}{2}h^2\lambda^3+\dfrac{1}{4}h^3\lambda^4)y_n \end{cases}$$

$$y_{n+1}=y_n+\frac{h}{6}(k_1+2k_2+2k_3+k_4)=[1+(h\lambda)+\frac{(h\lambda)^2}{2!}+\frac{(h\lambda)^3}{3!}+\frac{(h\lambda)^4}{4!}]y_n$$

相应的误差为: $\rho_{n+1}=[1+(h\lambda)+\dfrac{(h\lambda)^2}{2!}+\dfrac{(h\lambda)^3}{3!}+\dfrac{(h\lambda)^4}{4!}]\rho_n$

由稳定性要求 $|\rho_{n+1}| \le |\rho_n|$，于是得绝对稳定区域为：

$$\left|1+(h\lambda)+\frac{(h\lambda)^2}{2!}+\frac{(h\lambda)^3}{3!}+\frac{(h\lambda)^4}{4!}\right| \le 1$$

当λ为负实数时，可得绝对稳定区域：$-2.78 < \lambda h < 0$，因此，当λ的绝对值较大时，步长h限制很大，即h必须很小才能保证算法稳定性.

以例8为例求经典四阶R-K算法的绝对稳定区域

见Matlab程序
chapter2_example_8_RK.m
chapter2_example_8_stable_Euler_tri_RK.m

Matlab中的常微分方程求解指令

- ode45解非刚性微分方程,中等精度,使用Runge-Kutta法的四、五阶算法。
- ode23解非刚性微分方程，低精度,使用Runge-Kutta法的二、三阶算法。
- ode113解非刚性微分方程，Adams-Bashforth-Moulton PECE法。
- ode23t解中等的刚性微分方程，使用自由内插法的梯形法则。
- ode15s解刚性微分方程，使用可变阶次的数值微分(NDFs)算法。
- ode23s解刚性微分方程，低阶方法，使用修正的Rosenbrock公式。
- ode23tb解刚性微分方程，低阶方法，使用TR-BDF2方法.

注：常微分方程分为“刚性的”和“非刚性的”。刚性的是指其Jacobian矩阵的特征值相差十分悬殊。二者对步长选择的要求不同。

ode45的使用步骤：

（1）编写计算导数的M函数文件，该函数的输出量是“一阶导数向量Y’”，输入量是自变量和函数量Y；函数体描述一阶方程组右边的表达式。

（2）给定解方程的条件与要求

（3）调用指令求解并处理结果

ode45命令使用的语句格式如下：（其余指令用法类似）

[T,Y]=ode45(odefun,tspan,y0,options)，其中

odefun为要求解的常微分方程的函数句柄；
tspan为单调递增或递减的积分区间，如[1,8];
y0为初始条件矢量，其中矢量元素的排列顺序要求与函数中的元素顺序一致；

options为用odeset建立的优化选项，可以略去采用默认选项T是输出的时间列矢量，矩阵Y的每个列矢量是解的一个分量。

例6. 用Matlab自带指令求解下述常微分方程

$$\begin{cases} y' = y - \dfrac{2x}{y} & 0 \le x \le 1 \\ y(0) = 1 \end{cases}$$

的数值解，并与精确值比较。方程的解析解为:

$$y = \sqrt{1+2x}$$

见Matlab程序
chapter2_example_6_Matlab.m

• 例题7 用Matlab自带指令求解下述常微分方程

习题 2.5 下面两个耦合的一阶方程

$$\frac{dy}{dt}=p;\ \frac{dp}{dt}=-4\pi^2 y \tag{2.26}$$

确定了周期为1的简谐运动. 通过把上面给出的各个单变量公式之一推广到这个二变量情形, 以你选定的任何特定的初始条件积分方程组(2.26), 并且考察系统在 t 取整数值时回到其初始状态的精度.

见Matlab程序
chapter2_example_7_erbianliang_Matlab.m

例题7-2. 用Matlab自带指令求解下述常微分方程

$$\begin{cases} y''' - y'' = x \\ y(1) = 8 \\ y'(1) = 7 \\ y''(1) = 4 \end{cases}$$

其解析解为：

$$y = -\frac{x^3}{6} - \frac{x^2}{2} + 2.5x + 6e^{x-1} + \frac{1}{6}$$

对上述方程进行降阶，令 $y_1 = y,\ y_2 = y',\ y_3 = y''$

$$\begin{cases} y_1' = y_2 & y_1(1) = 8 \\ y_2' = y_3 & y_2(1) = 7 \\ y_3' = y_3 + x & y_3(1) = 4 \end{cases}$$

chapter2_example_7_2_sanbianliang_Matlab.m

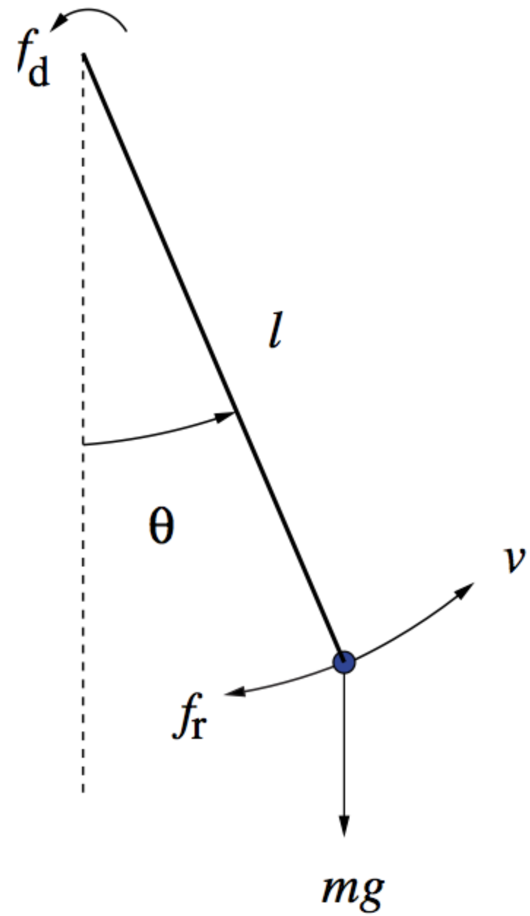
四、动力学中的有序和混沌

例 1. 强迫钟摆

一根长度为 l 的钟摆被限制在一垂直的平面内，在强迫外力 f_d 和阻力 f_r 的作用下振荡运动。钟摆的运动可以通过 Newton 方程来描述

$$ma = f_g + f_d + f_r$$

其中 $f_g = -mg\sin\theta$ 是重力在运动方向的分力，$a = l\, d^2\theta/dt^2$ 是沿切线方向的加速度，θ 是杆和垂线的夹角。



假设强迫外 力为 $f = f_d^0 cos\omega_0 t$

阻力为 $f_r = -kv = -kl\dfrac{d\theta}{dt}$

那么 Newton 运动方程就可以写成如下形式

$$\frac{d^2\theta}{dt^2} + q\frac{d\theta}{dt} + sin\theta = bcos\omega_0 t$$

其中 $q = \dfrac{k}{m}, b = \dfrac{f_d^0}{ml}$

微分方程降阶

并且取 $(l/g)^{1/2}$ 为时间单位。

设 $x_1 = \theta, x_2 = \frac{d\theta}{dt}$

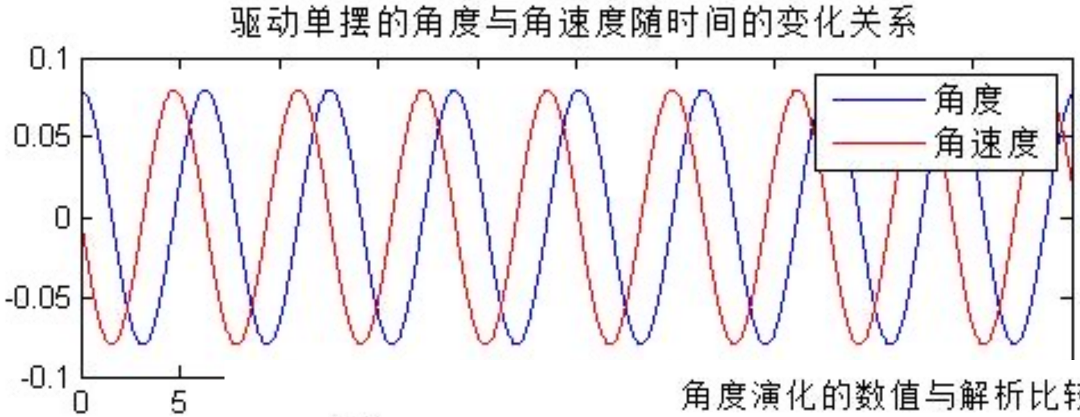
该运动方程就可以化为一阶方程组

$$\begin{cases} \frac{dx_1}{dt} = x_2 \\ \frac{dx_2}{dt} = -qx_2 - sinx_1 + bcos\omega_0 t \end{cases}$$

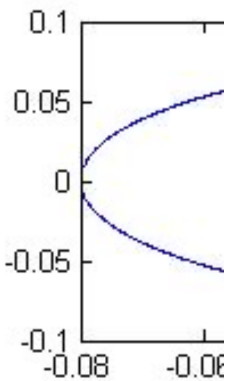
利用四阶RK算法求解上述方程组

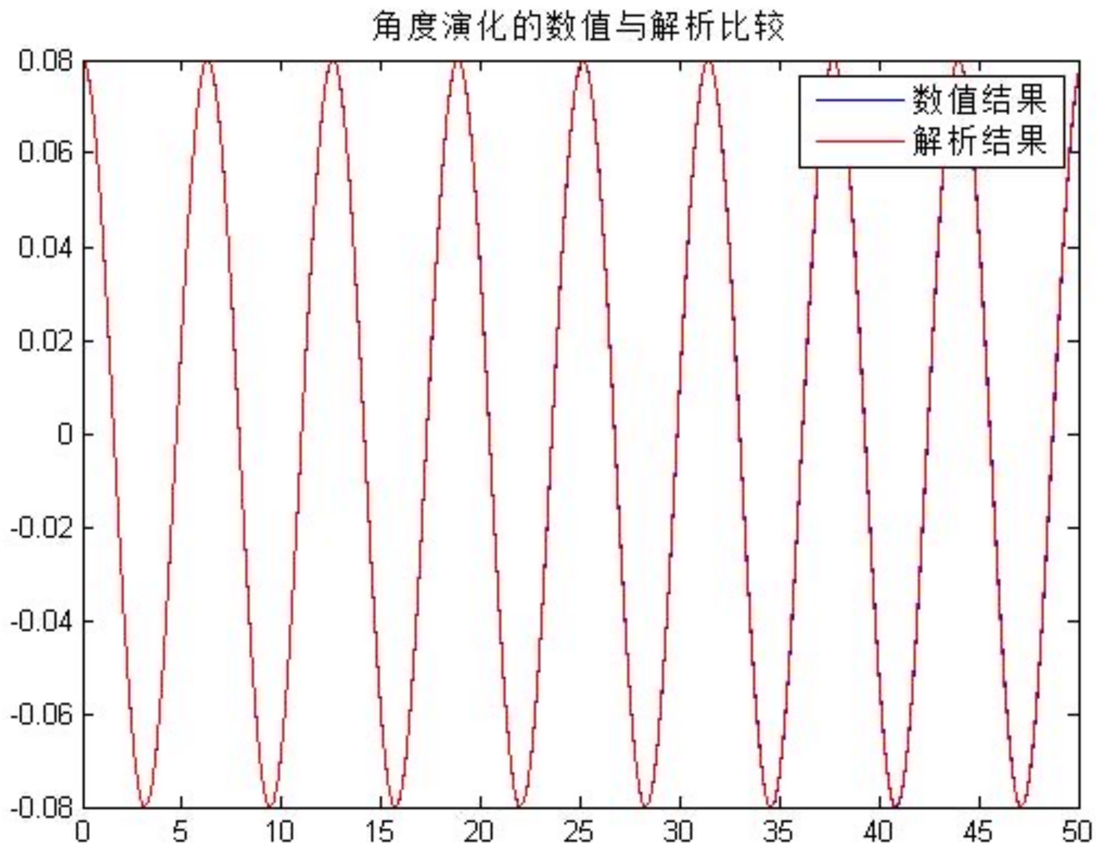
见Matlab程序
DrivenPendulum.m

见Matlab程序 DrivenPendulum_case1.m



杆和垂线的夹角、角速度随时间演化过程





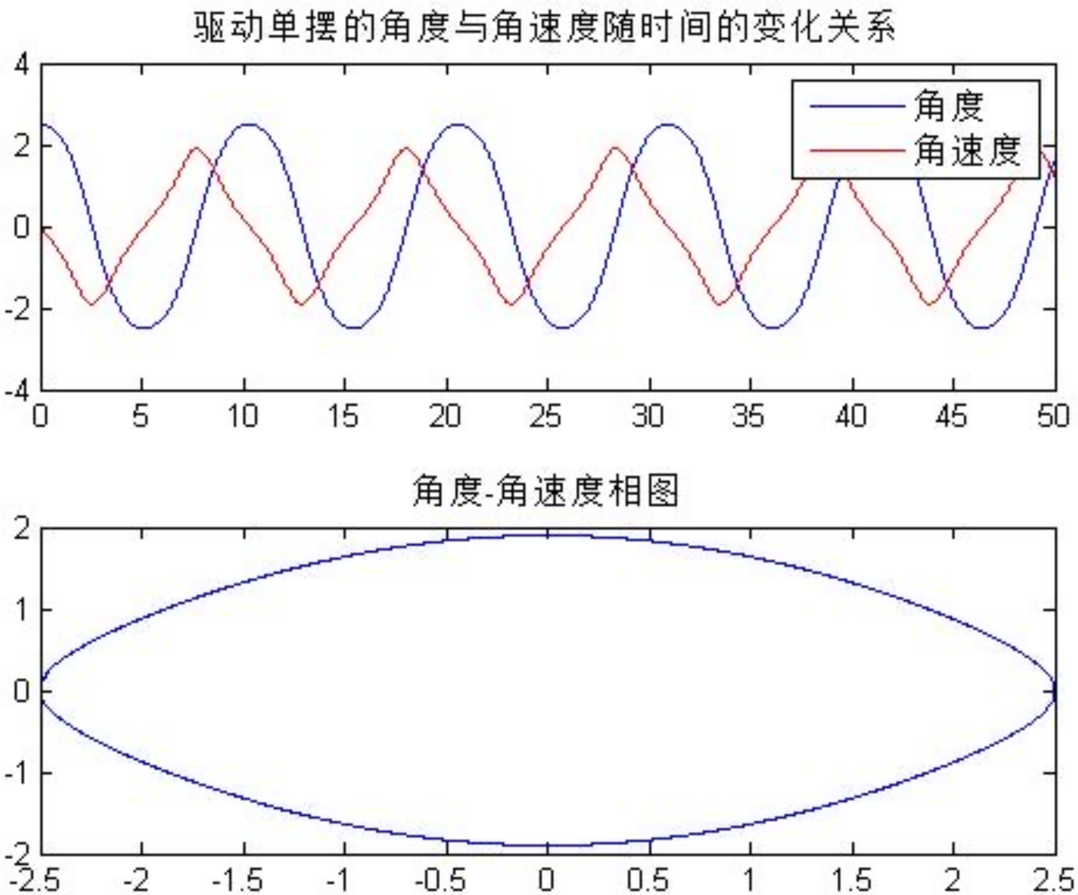
角的轨迹

q=0,b= 释放的

情况，钳 os *t*

与解析结

见Matlab程序 DrivenPendulum_case2.m



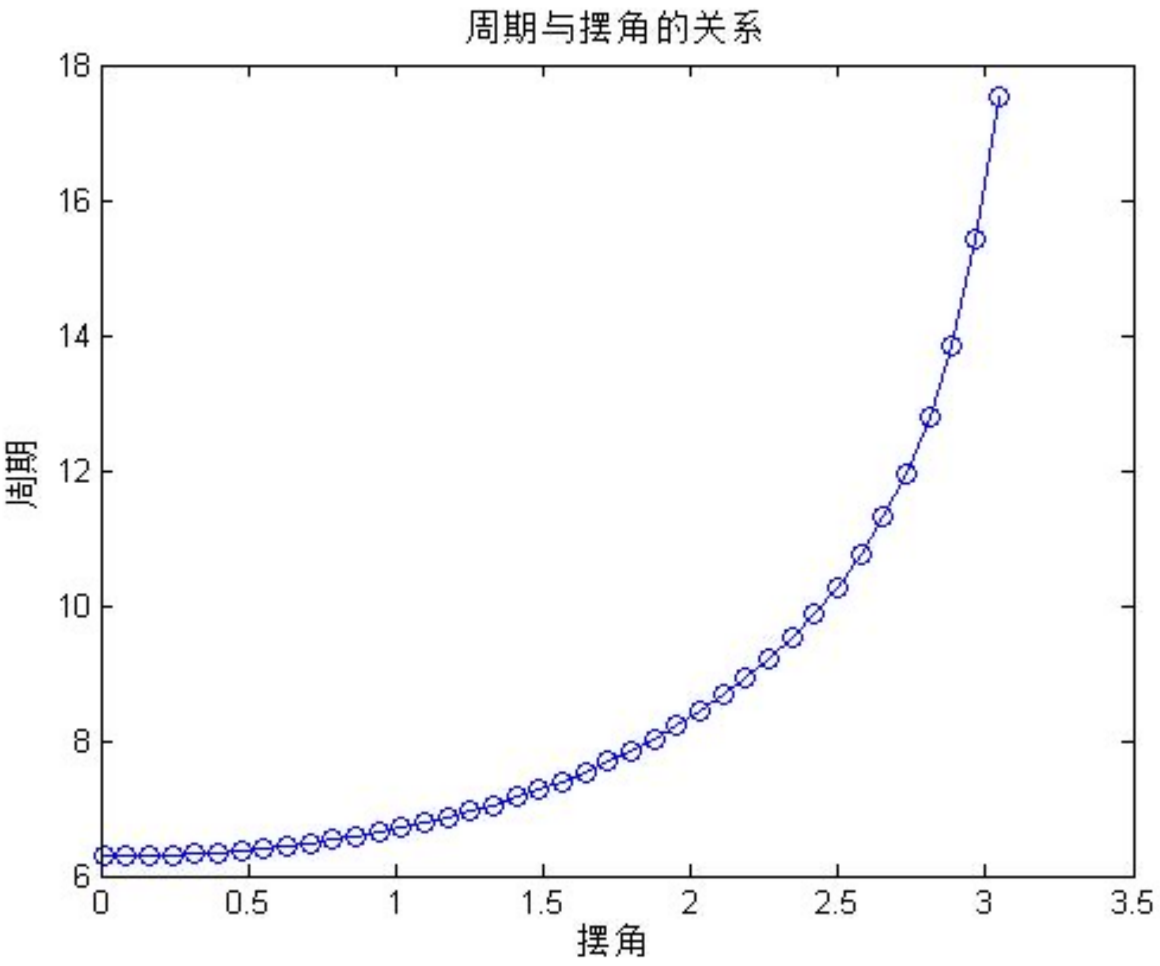
杆和垂线的夹角、角速度随时间演化过程

角速度与夹角的轨迹

q=0,b=0, t=50，初始角度为160°。即无驱动，无阻尼，大角度释放的情况。

显然钟摆运动不再是简谐运动,但仍是周期运动。

初始角度与周期的关系如何？

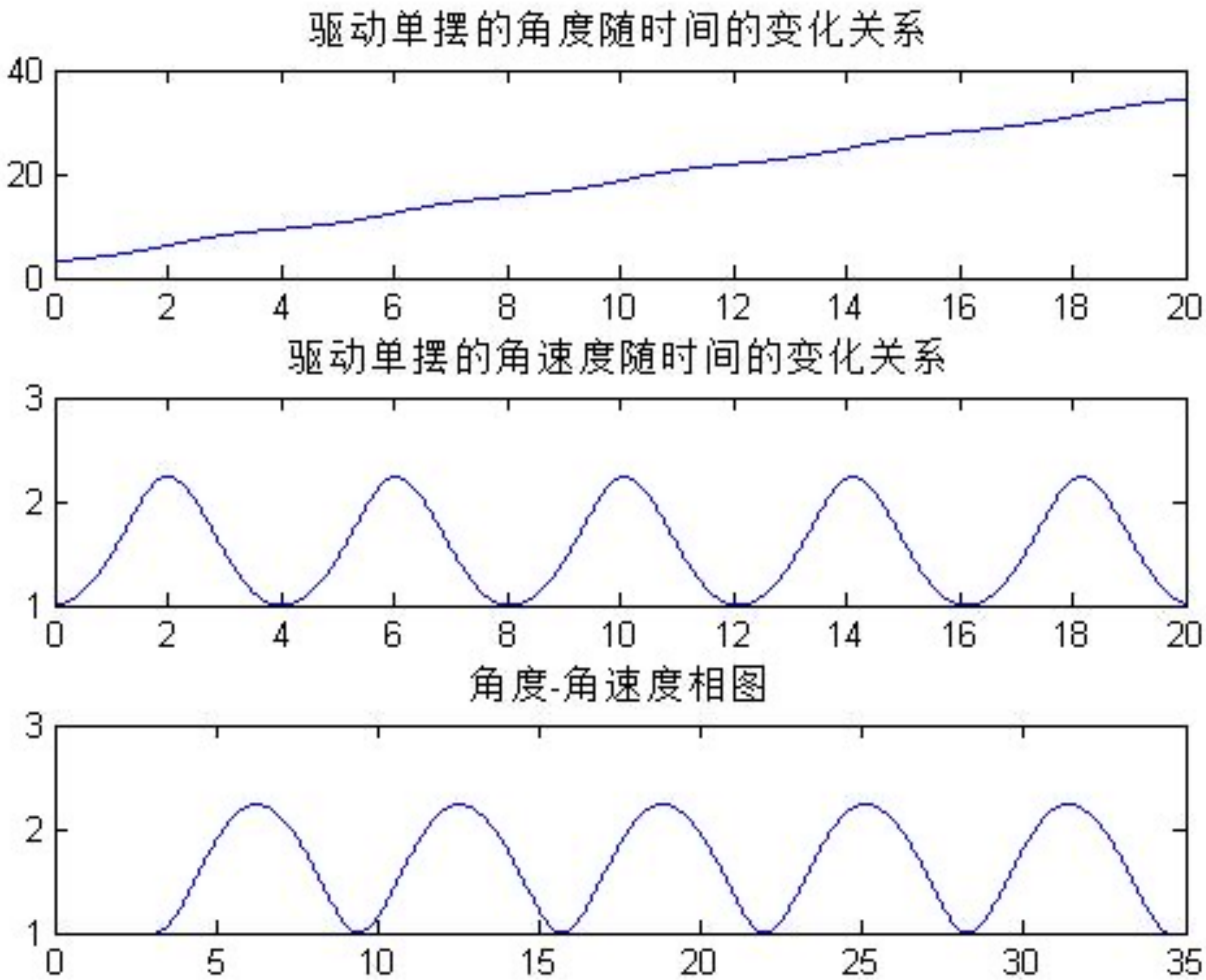


见Matlab程序

dirven_pendulum_duration.m

dirven_pendulum_duration2.m

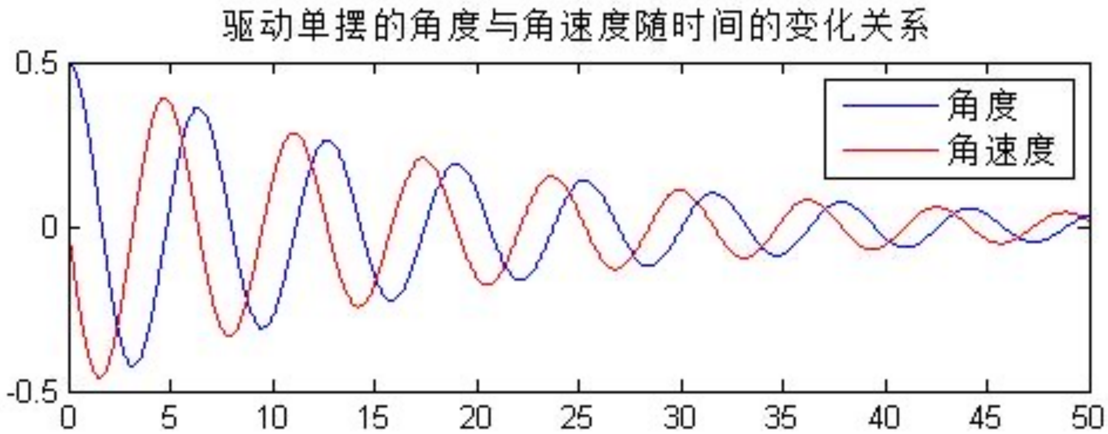
当初始摆角为180°时，单摆的运动如何?



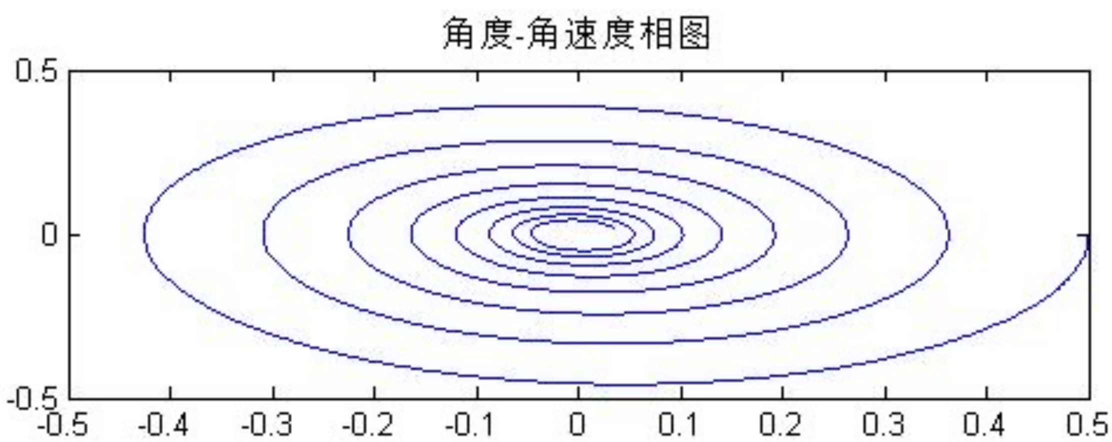
单摆做旋转运动，由初速度决定运动状态

见Matlab程序
DrivenPendulum_case3.m

见Matlab程序
DrivenPendulum_case4.m



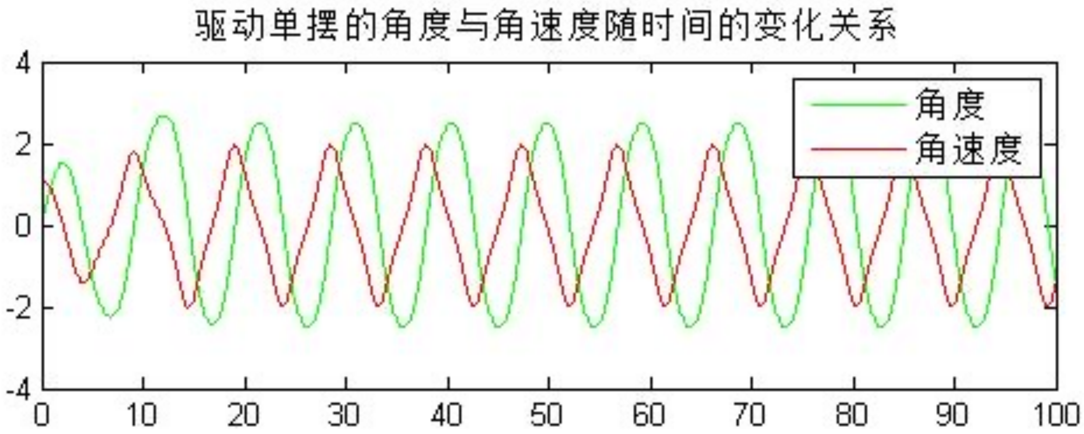
杆和垂线的夹角、角速度随时间演化过程



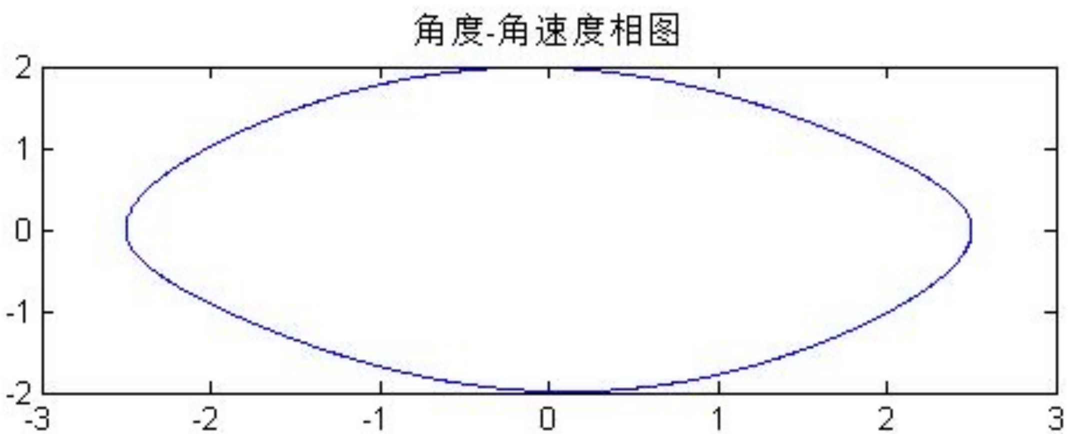
角速度与夹角的轨迹

q=0.2,b=0, ω_0 =2/3, t=50。即无驱动，有阻尼的情况。

在这种情况下,钟摆运动将逐步变慢直到停止。



杆和垂线的夹角、角速度随时间演化过程

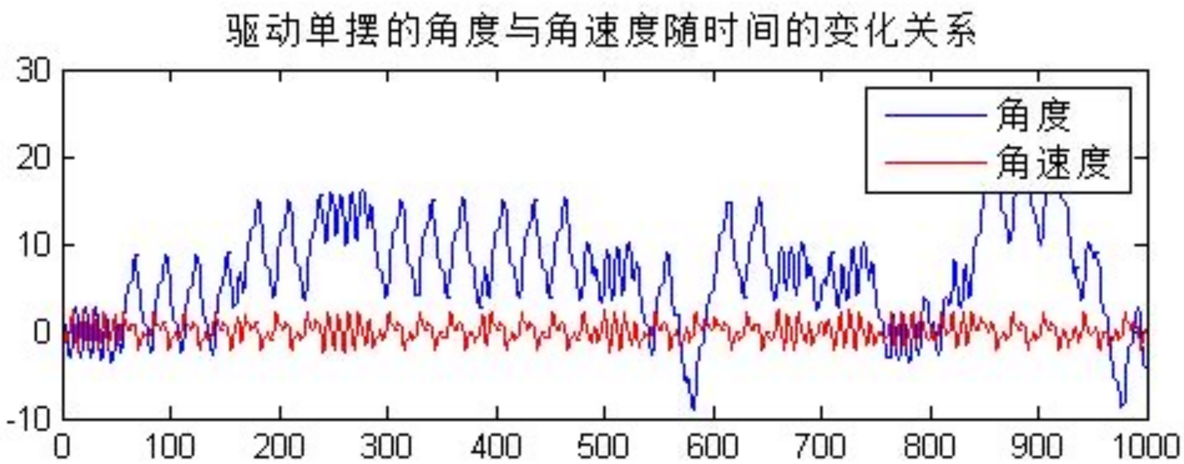


角速度与夹角的轨迹

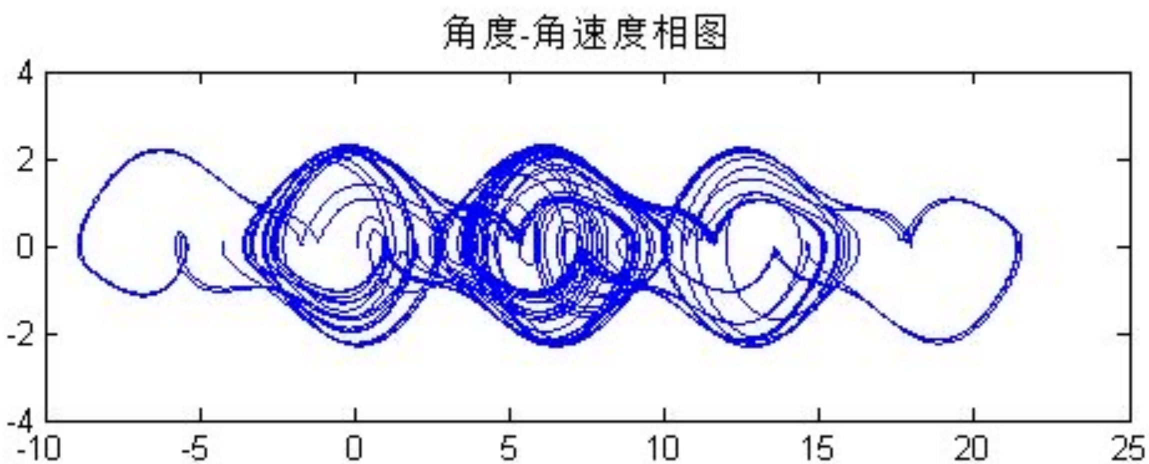
见Matlab程序 DrivenPendulum_case5.m

q=0.5,b=0.9, ω_0 =2/3, t=100。即小驱动，有阻尼的情况。

在这种情况下,钟摆运动将逐步稳定于一个有序的周期运动



杆和垂线的夹角、角速度随时间演化过程



见Matlab程序 DrivenPendulum_case6.m

角速度与夹角的轨迹

$q=0.5$, $b=1.15$, $\omega_0=2/3$, $t=1000$，即大驱动，有阻尼的情况。

在这种情况下钟摆运动是一个无序运动,呈现出分形的特征

无阻尼，大驱动情况的结果如何？

例题2. 简谐振子的运动方程为 $m\dfrac{\mathrm{d}^2x}{\mathrm{d}t^2}=-kx$

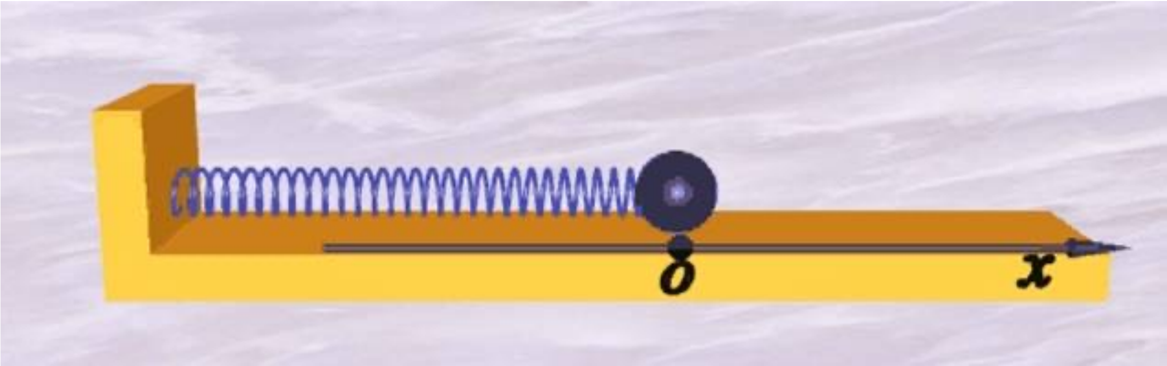
初始条件： $x_0=5;\ p_0=0$

其精确解为： $x=A\cos(w_0t+\phi)=A\cos(\sqrt{\dfrac{k}{m}}t+\phi)$

$$p=-mAw_0\sin(w_0t+\phi)$$

其中振幅A与初始相位由初始条件决定。

$$\frac{1}{2}kx_0^2+\frac{1}{2}mv_0^2=\frac{1}{2}kA^2 \qquad \phi=\cos^{-1}(x_0/A)$$



令 $m=1, k=-4\pi^2$

采用降阶处理，运动方程化为：

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = p/m = p \\ \dfrac{\mathrm{d}p}{\mathrm{d}t} = -kx = -4\pi^2 x \end{cases}$$

上述方程对应于一个周期为1的简谐振动。

利用四阶Runge-Kutta算法可以数值求解运动方程

见Matlab程序

chapter2_harmonic_oscillator_RK.m

例3. 简谐振子的受迫振动

无阻尼情况： $$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = -kx + F\cos\omega_\mathrm{p}t$$

驱动力

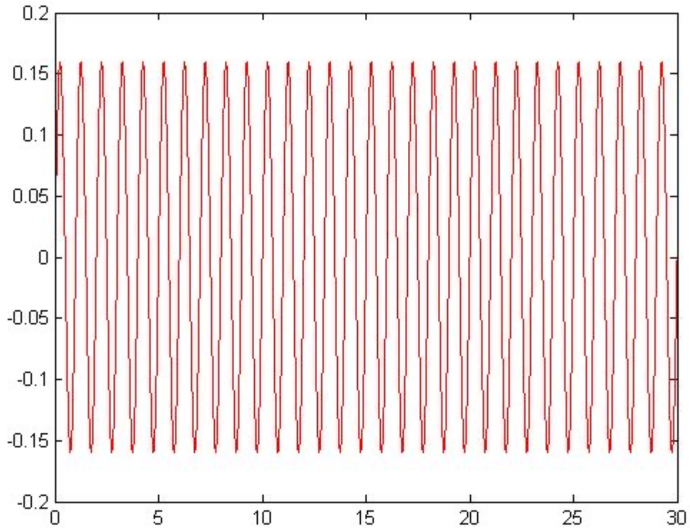
令 $\omega_0 = \sqrt{\dfrac{k}{m}}$ $f = F/m$

可得： $$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \omega_0^2x = f\cos\omega_\mathrm{p}t$$

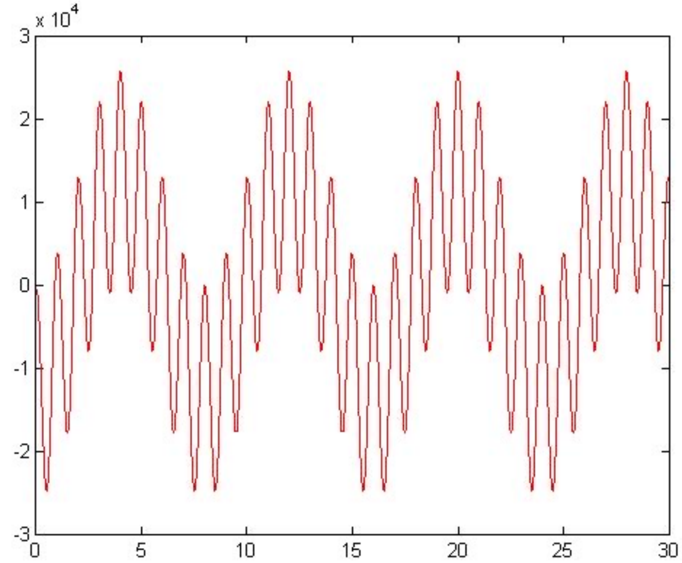
其精确解为： $x = A\cos(\omega_0t + \phi) + A'\cos\omega_\mathrm{p}t$

其中： $$A' = \frac{f}{\omega_0^2 - \omega_\mathrm{p}^2}$$

无阻尼时



驱动力为零时的**X(t)**曲线



驱动力不为零时的**X(t)**曲线

驱动力不为零时的运动并非简谐振动！为什么？

见Matlab程序

chapter2_driven_harmonic_oscillator_RK.m

例3. 简谐振子的受迫振动

阻尼项

有阻尼情况： $m\dfrac{\mathrm{d}^2x}{\mathrm{d}t^2}+C\dfrac{\mathrm{d}x}{\mathrm{d}t}+kx=F\cos\omega_{\mathrm{p}}t$

驱动力

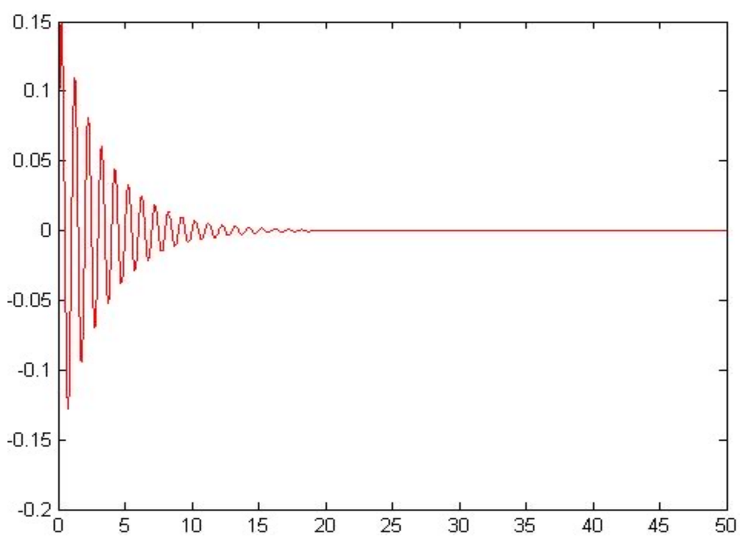
令 $\omega_0=\sqrt{\dfrac{k}{m}}\qquad 2\delta=C/m\qquad f=F/m$

可得： $\dfrac{\mathrm{d}^2x}{\mathrm{d}t^2}+2\delta\dfrac{\mathrm{d}x}{\mathrm{d}t}+\omega_0^2x=f\cos\omega_{\mathrm{p}}t$

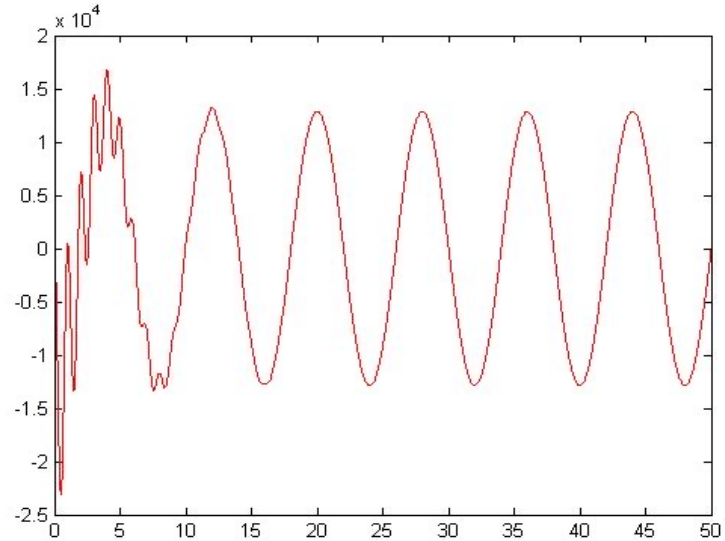
解析解为： $x=\cancel{A\,\mathrm{e}^{-\delta t}\cos(\omega t+\phi)}+A'\cos(\omega_{\mathrm{p}}t+\psi)$

其中： $A'=\dfrac{f}{\sqrt{(\omega_0^2-\omega_{\mathrm{p}}^2)^2+4\delta^2\omega_{\mathrm{p}}^2}},\ \tan\psi=\dfrac{-2\delta\omega_{\mathrm{p}}}{\omega_0^2-\omega_{\mathrm{p}}^2}$

有阻尼时



驱动力为零时的**X(t)**曲线



驱动力不为零时的**X(t)**曲线

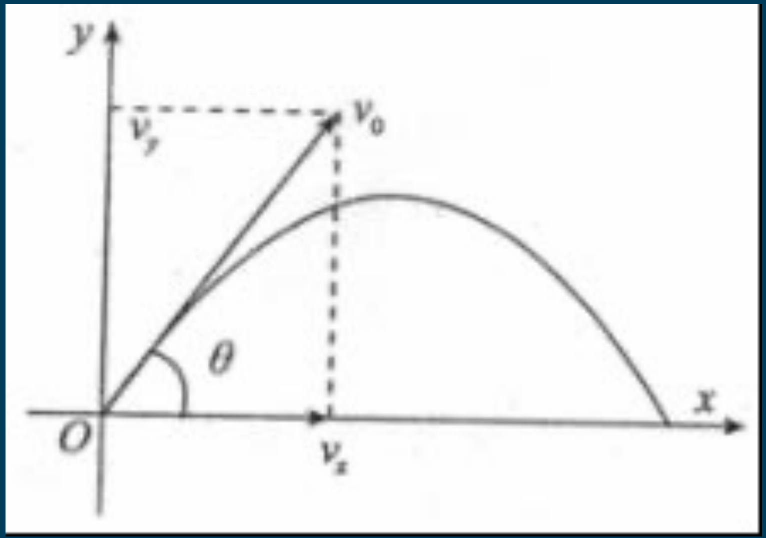
有阻尼情况下，驱动力频率为简谐运动频率！

见Matlab程序

chapter2_driven_harmonic_oscillator_RK_damp.m

第二次作业　阻尼斜抛运动的数值求解

研究受空气阻力作用的斜抛运动。设抛体质量为m，初速度为v_0，受空气阻力的大小R与速度v的n次方成正比，即$R=bv^n$，其中b是阻尼系数。



将抛体视为质点，根据牛顿运动定律，抛体的运动微分方程可以写为

$$m\frac{d^2\vec{r}}{dt^2} = -m\vec{g} - bv^n\frac{\vec{v}}{v} = -m\vec{g} - bv^{n-1}\vec{v}$$

以抛出点为原点建立直角坐标系，Ox沿水平方向，Oy沿垂直向上。则上述运动方程可以化为

$$\begin{cases} \dfrac{d^2x}{dt^2} = -\dfrac{b}{m}(v_x^2 + v_y^2)^{\frac{n-1}{2}} v_x \\ \dfrac{d^2y}{dt^2} = -g - \dfrac{b}{m}(v_x^2 + v_y^2)^{\frac{n-1}{2}} v_y \end{cases}$$

对上述方程组进行降阶，令

$$y_1 = x; y_2 = \frac{dx}{dt} = v_x; y_3 = y; y_4 = \frac{dy}{dt} = v_y$$

可得：

$$\begin{cases} \dfrac{dy_1}{dt} = y_2; \dfrac{dy_2}{dt} = -\dfrac{b}{m}(y_2^2 + y_4^2)^{\frac{n-1}{2}} y_2 \\ \dfrac{dy_3}{dt} = y_4; \dfrac{dy_4}{dt} = -g - \dfrac{b}{m}(y_2^2 + y_4^2)^{\frac{n-1}{2}} y_4 \end{cases}$$

令初始条件为：

$$y_1 = 0; y_2 = \frac{dx}{dt} = 30; y_3 = 0; y_4 = \frac{dy}{dt} = 50$$

求以下情况中抛体的运动轨迹

（1）b=0，即没有阻尼的情形；

（2）有阻尼，n=1的情形；

（3）有阻尼，n=2的情形；