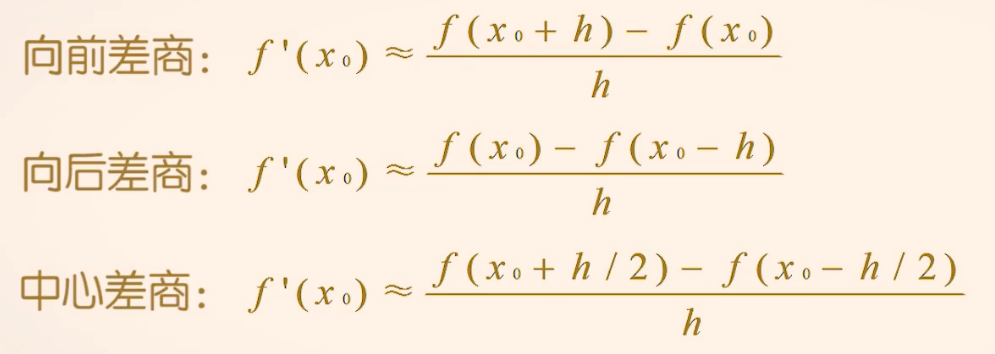
# 数值微积分与方程求解

所有题目源码见untitled.m文件

## 数值微分与数值积分

### 数值微分

1. 数值差分与差商

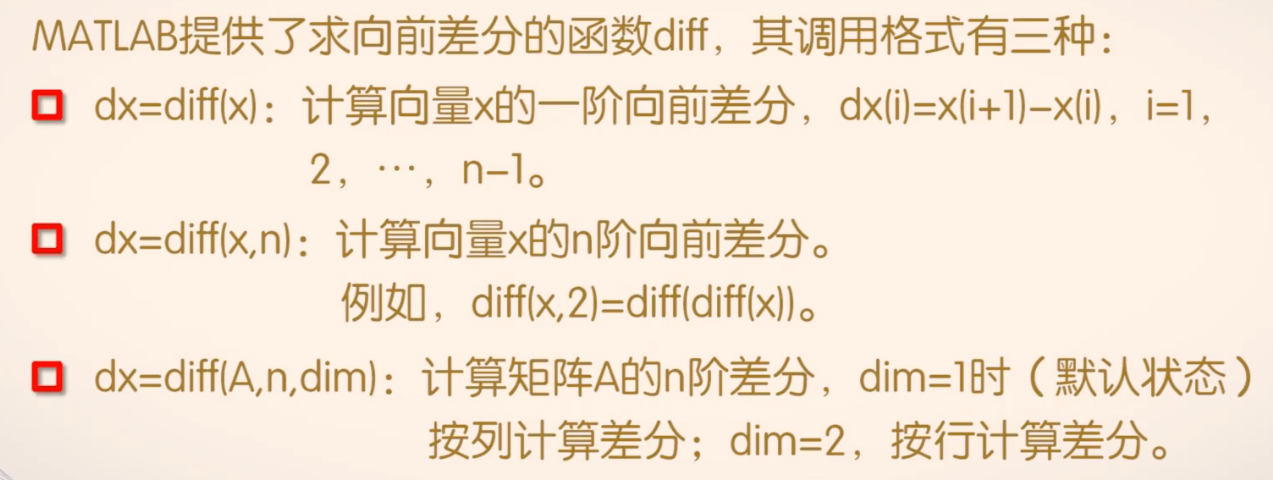


导数是通过极限来定义的

当步长h足够小时，函数在x0点处以h为步长的三种差商公式，函数f(x)在x0点的微分接近于函数在该点的差分，而f(x)在x0点的导数接近于函数在该点的差商。

1. 数值微分的实现

diff()函数，调用格式有三种



注意新生成的矩阵元素比原来小1个范围，n个元素只有n-1个空，可以用这个计算导数的近似值。

### 数值积分

**基于自适应辛普森方法、基于自适应Gauss-Lobatto方法：**

1．[I,n]=quad(filename,a,b,tol,trace)

2．[I,n]=quadl(filename,a,b,tol,trace) %效率和精度比上面都要高一点

第一个参数为被积函数名，a、b（必须是有限的）代表积分上下限，tol用于控制积分精度，默认10-6，trace默认为0，不展现积分精度，n为被积函数的调用次数。

**基于全局自适应积分方法：**

i=integral(filename,a,b)

与上式类似，a、b可以是无限的。

**基于自适应高斯-克朗罗德方法：**

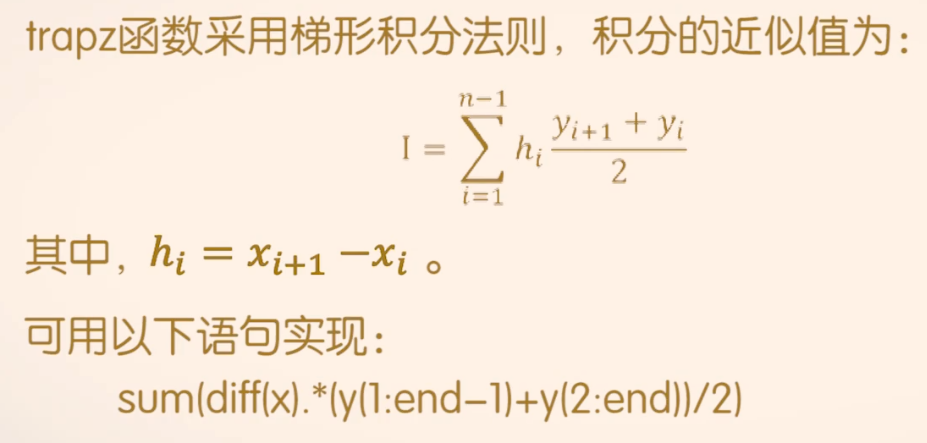
[I,err]=quadgk(filename,a,b)

其中，err返回近似误差范围，积分上下限可以无穷或复数。

**基于梯形积分法：**

i=trapz(x,y)

当被积函数未知时用这个求，其中定义了 y=f(x)函数关系，x,y向量等长，且x递增。



### 多重定积分的数值求解

求二重积分：

i=integral2(filename,a,b,c,d)

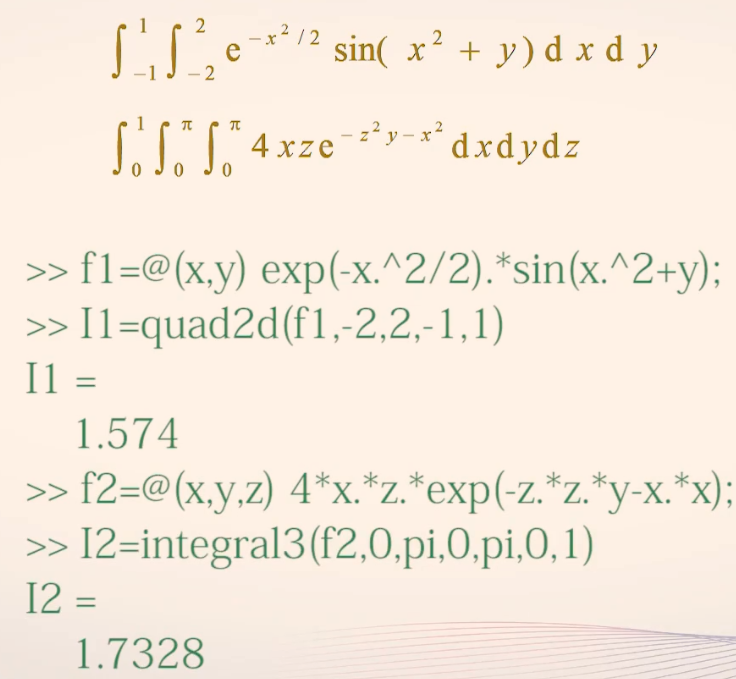
i=quad2d(filename,a,b,c,d)

i=triplequad(filename,a,b,c,d,tol)

求三重积分：

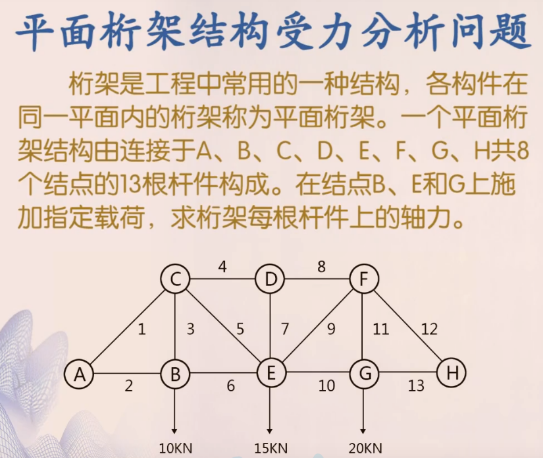
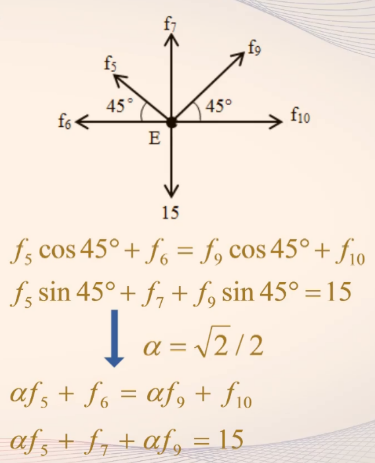
i=integral3(filename,a,b,c,d,e,f)

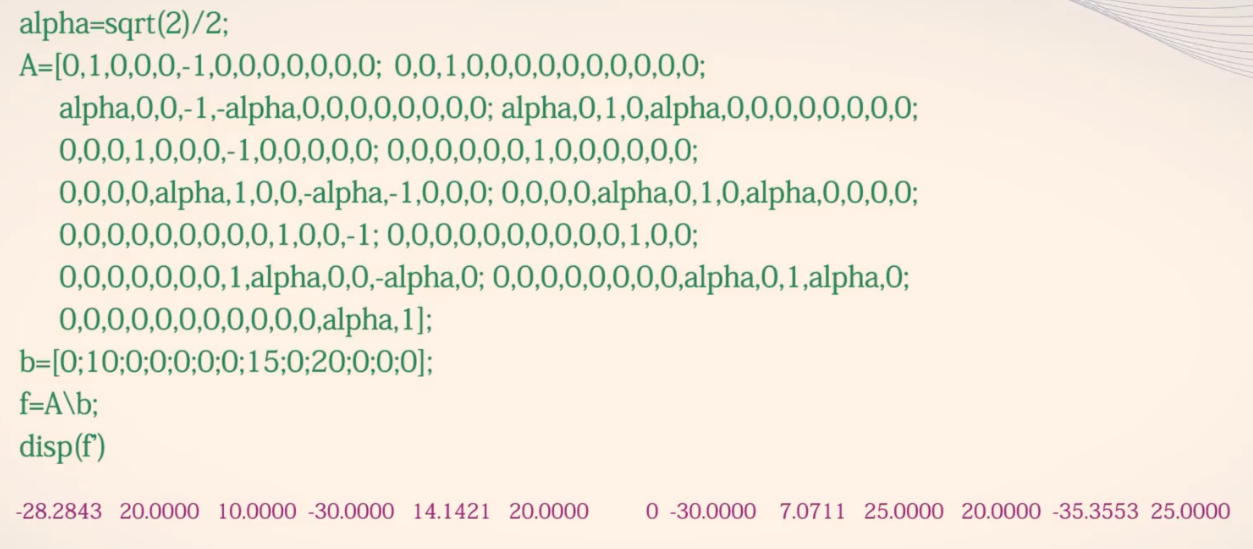
i=triplequad(filename,a,b,c,d,e,f,tol)

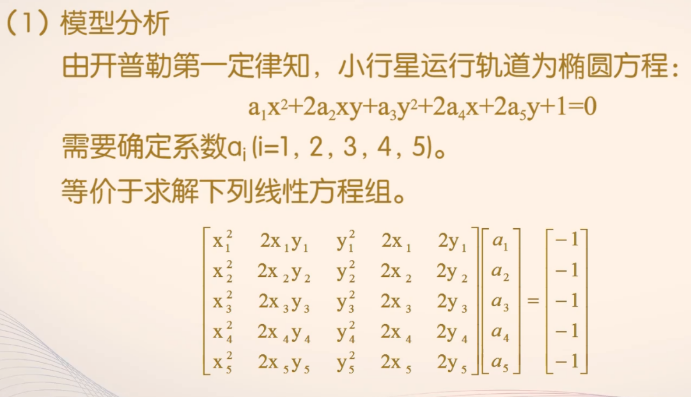


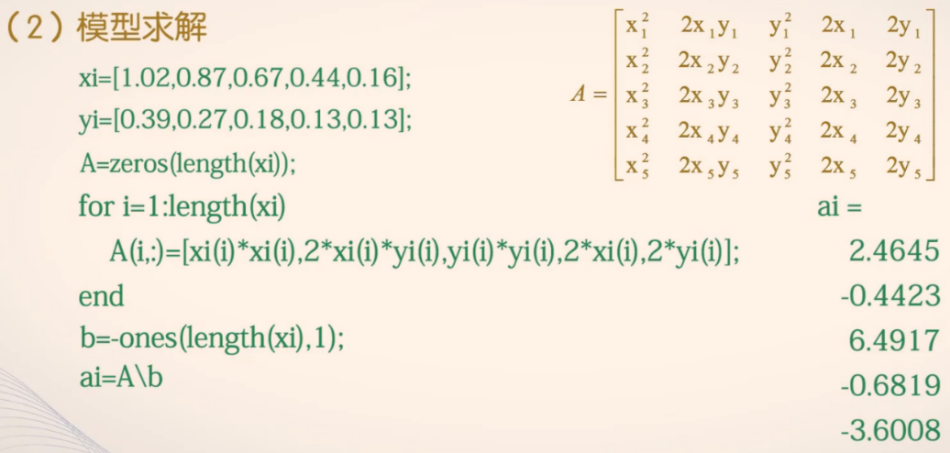
## 线性方程组应用举例

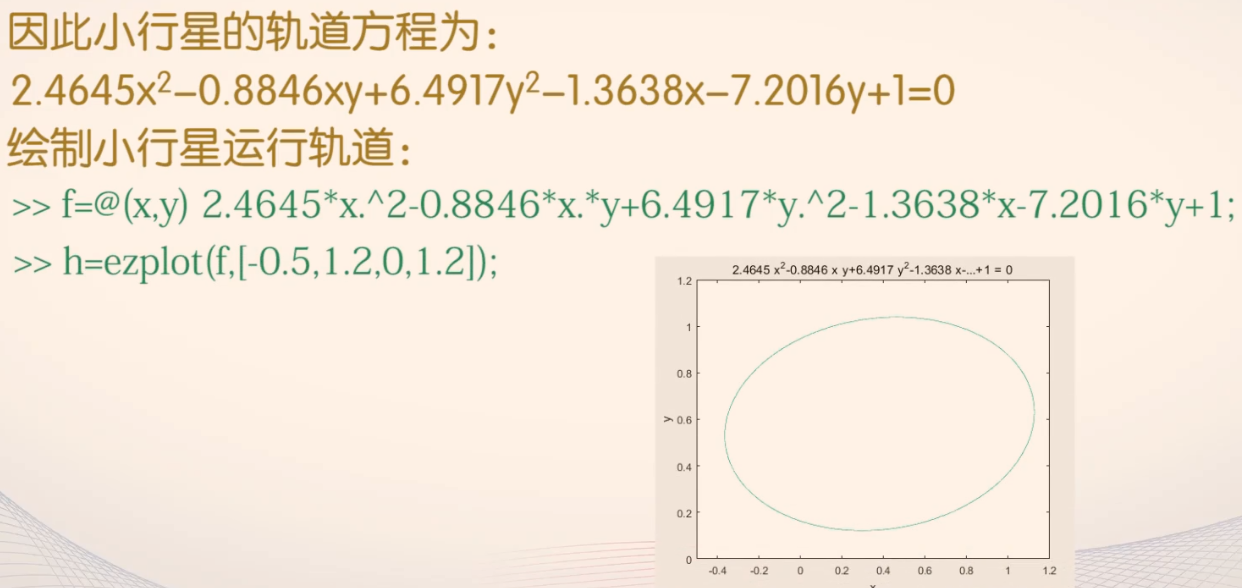
见脚本文件







## 非线性方程求解与函数极值计算

### 非线性方程数值求解

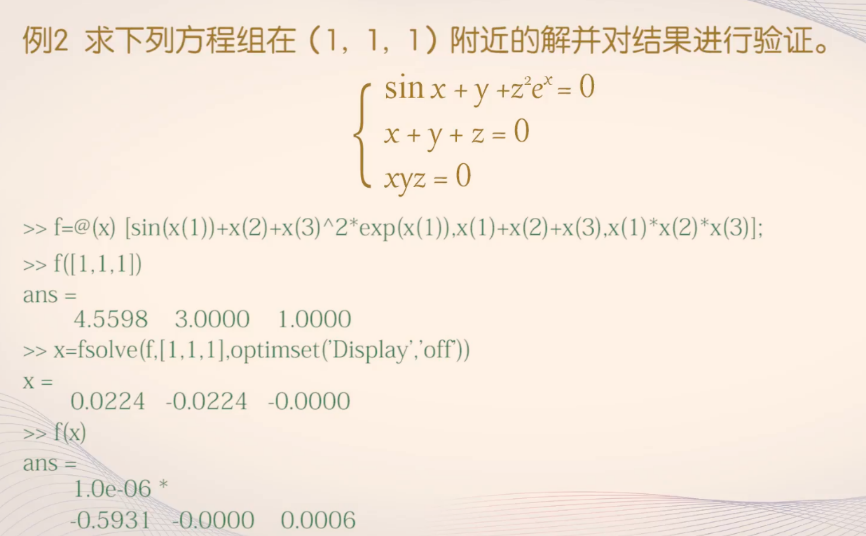
1. 牛顿迭代法

**x=fzero(filename,x0)**

其中，filename是待求方程左端的函数表达式，x0是初始值，初值的选取对结果影响很大，实际上是一个搜索函数。

**x=fsolve(filename,x0,option) % 功能更强**

其中，x为返回的近似解，filename是待求方程左端的函数表达式，x0是初始值，option用于设置优化工具箱的优化参数，可以调用optimset函数来完成。



1. 二分法
2. 割线法

### 函数极值计算

MATLAB只考虑函数极小值的计算，极大值可以求-f(x)的极小值。

求最小值的函数有三种：

[xmin,fmin]=fminbnd(filename,x1,x2,option) % 适用于一元函数

[xmin,fmin]=fminsearch(filename,x0,option) % 求多元函数

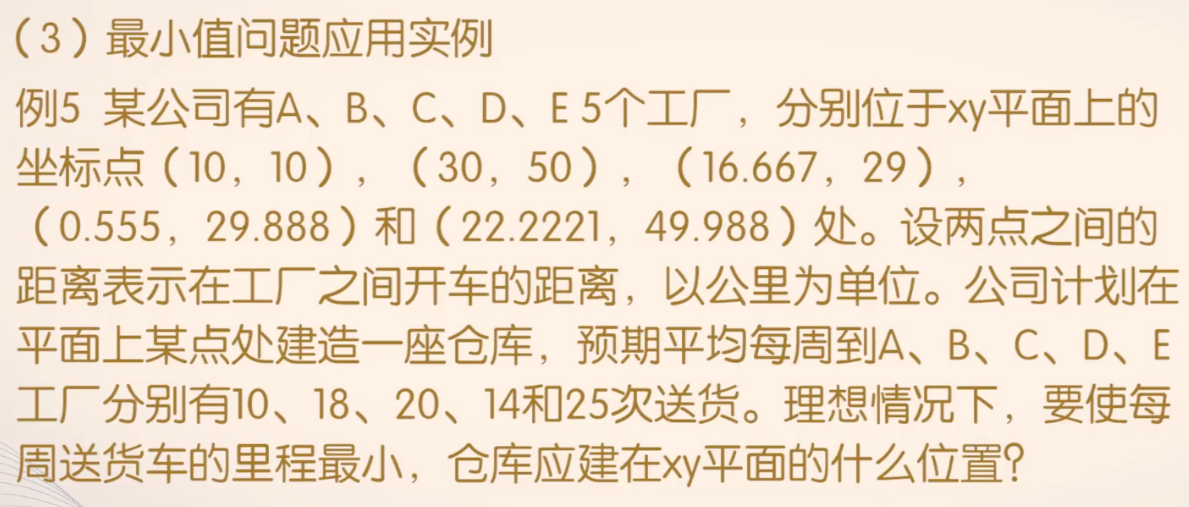
[xmin,fmin]=fminunc(filename,x0,option) % 求多元函数，推荐这个

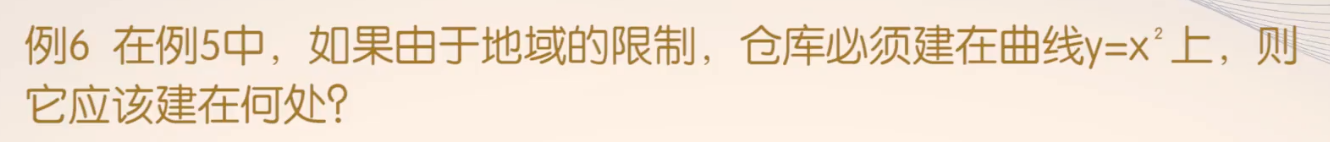
有约束最优化问题：

[xmin,fmin]=fmincon(filename,x0,A,b,Aep,bep,Lbnd,Ubnd,NonF,option)

参数实在太多了，用的时候查帮助文档吧

最小值问题实例：

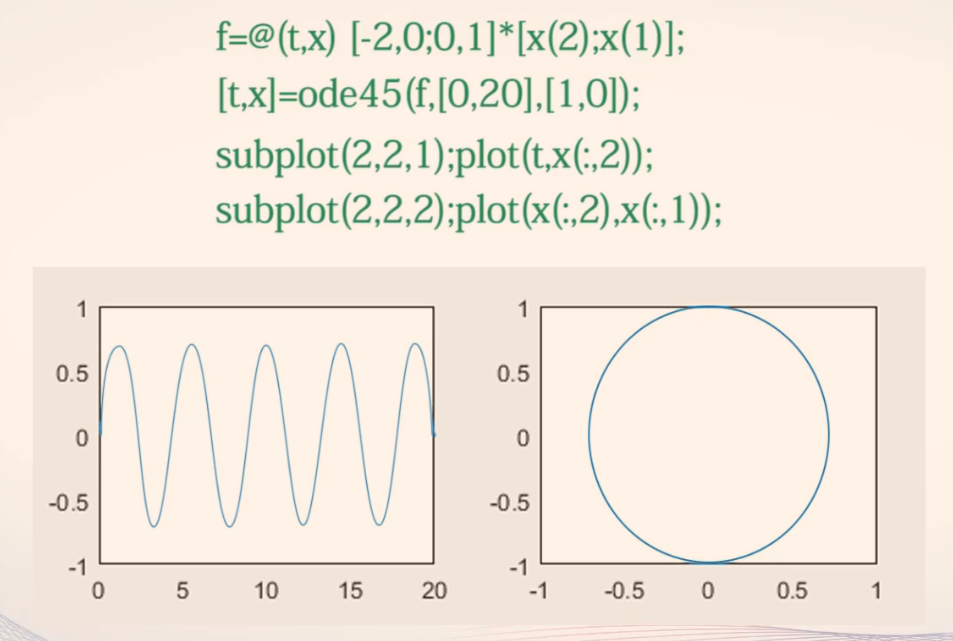


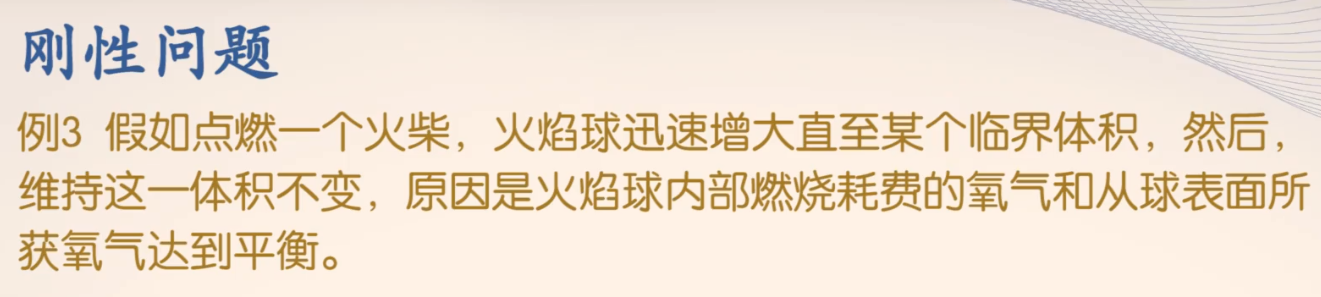


## 常微分方程数值求解

常微分方程统一求解格式：**odennxx**

其中ode代表常微分方程的意思；nn是数字，代表所用方法的阶数；xx是字母，用于标注方法的专门特征。





## 常微分方程应用举例