第三节 微分方程

微分方程是自然科学和社会科学研究中非常重要且十分实用的一种数学学科.本实验的目的是学会使用数学软件求解一般常见的常微分方程和微分方程组.常微分方程有解析解法和数值解法两种.解析解法是通过解表达式能让我们更清楚地理解解的性态,但其求解过程不规范而且十分复杂,因而只能对某些特殊类型的方程求解,对于多数方程则不易或不能求解;而数值解法的求解已有一些常用方法,如 Eular 法、Runge-Kutta 法等,它们能求出一般常微分方程在某一指定区间满足一定精度的数值解.

一、常微分方程的通解

当微分方程中不含任何附加条件(初值条件)时,其解就是该方程的通解,通解中任意常数的个数(一般要求独立)等于微分方程的阶数.在 MathCAD 中没有求微分方程解析解的方法.在 MATLAB 中,可用函数 dsolve( )求解一般微分方程,包括可分离变量微分方程、齐次方程、一阶、二阶、n 阶线性齐次和非齐次微分方程、可降阶的高阶微分方程等.例如

```
syms x y z t a b
dsolve('Dy-x*Dy=a*(y^2+Dy)','x')
ans =1/(log(-1+x+a)*a+C1)
dsolve('sin(x)*cos(y)*Dy+cos(x)*sin(y)=0','x')
ans =asin(exp(C1)/sin(x))
dsolve('2*exp(x/y)*(1-x/y)*Dy+(1+2*exp(x/y))=0','x')
ans = -x/lambertw(2*x/(x-C1))
dsolve('t*Dy+y=t^2+3*t+2')
ans =1/6*(2*t^3+9*t^2+12*t+6*C1)/t
dsolve('D3y=exp(2*x)-cos(x)','x')
ans =1/8*exp(2*x)+sin(x)+C1+C2*x+C3*x^2
dsolve('(1+x^2)*D2y=2*x','x')
ans = -2*x+2*atan(x)+log(1+x^2)*x+C1+C2*x
dsolve('y*D2y-(Dy)^2=0','x')
```

```
ans  = exp(C1 * x + C2 * C1)
dsolve('x * D2y + Dy = 0','x')
ans  = C1 + C2 * log(x)
dsolve('D2y - 2 * Dy + y = 0','x')
ans  = C1 * exp(x) + C2 * exp(x) * x
dsolve('D2y - 2 * Dy + y = exp(x)/x','x')
ans  = - exp(x) * x + log(x) * exp(x) * x + C1 * exp(x) + C2 * exp(x) * x
dsolve('D2y - 2 * Dy + 5 * y = 0','x')
ans  = C1 * exp(x) * cos(2 * x) + C2 * exp(x) * sin(2 * x)
dsolve('D2y - 2 * Dy + 5 * y = exp(x) * sin(2 * x)','x')
ans  = - 1/4 * exp(x) * cos(2 * x) * x + 1/16 * exp(x) * cos(2 * x) * sin(4 * x) - 1/16
 * cos(4 * x)  * exp(x) * sin(2 * x) + C1 * exp(x) * cos(2 * x) + C2 * exp(x) *
sin(2 * x)
dsolve('D2y - y = sin(x)^2','x')
ans  = 1/5 * ( - 3 * exp(x) + cos(x)^2 * exp(x) + 5 * C1 * exp(x)^2 + 5 * C2)/exp(x)
```

用 dsolve( )函数求解微分方程通解的一般形式为

dsolve('微分方程','自变量')

其中“微分方程”和“自变量”都按字符串形式给出,并用单引号括住,且当自变量为 t 时可省略.这里微分方程的写法不同于常规写法:微分算子用“D”表示(D 必须大写),用 Dny 表示 $y^{(n)}$,即 y 对自变量的 n 阶导数,n 可以是任何已知的常数.如 D5y 表示 y 的五阶导数.

值得一提的是,用 dsolve( )函数求出的微分方程的解的一般形式均为 $f(x)$,即解可以表示为 $y=f(x)$的形式.如果微分方程的解为隐函数,即不能表示成显式解,则 MATLAB 用兰伯特的 W 函数 Lambertw( )表示微分方程的解,如上述求

$$2\exp\left(\frac{x}{y}\right)\left(1-\frac{x}{y}\right)\mathrm{d}y+\left(1+2\exp\left(\frac{x}{y}\right)\right)\mathrm{d}x=0$$

的解,其结果为

$$y=-\frac{x}{\mathrm{Lambertw}\left(\dfrac{2x}{x-C1}\right)},$$

但若将求出的解用隐函数表示,则其解为

$$x+2y\exp\left(\frac{x}{y}\right)=C1,$$

dsolve( )函数在通解中自动用 C1,C2,…表示任意常数.

二、常微分方程的特解

给定微分方程及其初值条件,求微分方程满足一定初值条件的特解是解微分方程的基本内容.在 MATLAB 中仍用函数 dsolve( )来求微分方程的特解,其一般形式为

dsolve('微分方程','初值条件 1,条件 2,…','自变量')

其中微分方程与自变量同前所述,书写规则也一样,初值条件可以有多个,用逗号隔开.如有初值条件:$y|_{x=x_0}=y_0, y'|_{x=x_0}=y_1$,则初值条件的写法为' y(x0) = y0, Dy(x0) = y1 ' ,其它情况可依此类推.例如

```
syms x y a b t
dsolve('Dx = a * x * (b - x)','x(0) = x0')
ans = b/(1 + exp( - a * b * t) * (b - x0)/x0)
dsolve('y + (x + y) * Dy = 0','y(0) = 0','x')
ans = [ - x + (x^2)^(1/2)]
      [ - x - (x^2)^(1/2)]
dsolve('y * Dx + x + y = 0','y')
ans = - 1/2 * (y^2 - 2 * C1)/y
dsolve('Dy - y/(x + 2) = x^2 + 2 * x','y( - 1) = 3/2','x')
ans = 1/2 * x^3 + x + x^2 + 2
dsolve('Dy = (y/x)^2 + y/x + 4','y(1) = 0','x')
ans = 2 * tan(2 * log(x)) * x
dsolve('x * D2y + Dy = 0','y(1) = 1, Dy(1) = 6','x')
ans = 1 + 6 * log(x)
dsolve('D2y - 10 * Dy + 25y = 0','y(0) = 0, Dy(0) = 1','x')
ans = 5/2 * x + 3/20 - 3/20 * exp(10 * x)
dsolve('D2y - 8 * Dy + 7y = 37 * exp(x) * sin(x)','y(0) = 0, Dy(0) = 1','x')
ans = 7/8 * exp( - 8 * x) * x * exp(8 * x) + 111/50 * exp( - 8 * x) * exp(9 * x)
       * cos(x) - 74/25 * exp( - 8 * x)  * exp(9 * x) * sin(x) + 7/64 * exp( - 8
       * x) * exp(8 * x) - 39/16 + 173/1600 * exp(8 * x)
dsolve('D4y - y = 2 * exp(x)','y(0) = 1, Dy(0) = 1, D2y(0) = 1, D3y(0) = 1','x')
ans = 1/4 * (2 * x * exp(x)^2 + exp(x)^2 + 2 * exp(x) * cos(x) + 2 * exp(x) *
      sin(x) + 1)/exp(x)
dsolve('D3y - 2 * D2y - 3 * Dy = 0','y(0) = 0, Dy(0) = 1, D2y(0) = 2','x')
ans = ( - 1/4 + 1/4 * exp(3 * x) * exp(x))/exp(x)
dsolve('y^3 * D2y + 1 = 0','y(1) = 1, Dy(1) = 0','x')
```

```
ans = (-x^2+2*x)^(1/2)
dsolve('D3y=exp(a*x)','y(1)=0,Dy(1)=0,D2y(1)=0','x')
ans = (exp(a*x)-1/2*exp(a)*(2+a^2-2*a)+exp(a)*(-1+a)*a*x
      -1/2*a^2*exp(a)*x^2)/a^3
dsolve('D2y+(Dy)^2=1','y(0)=0,Dy(0)=0','x')
ans = [ log(cosh(x)*cosh(i*pi)-sinh(x)*sinh(i*pi))-i*pi]
      [ log(cosh(x)*cosh(0)-sinh(x)*sinh(0))]
dsolve('D2y-a*(Dy)^2=0','y(0)=0,Dy(0)=1','x')
ans = (-log(a*x-1)+i*pi)/a
```

由上述结果可以看出，当微分方程的解可以显式表示时，给出微分方程的显式解；若微分方程的显式解不唯一时，则同时给出各个显式解．但当微分方程的解不能显式表示时，则要么写不出解的表达式，要么就用兰伯特的 W 函数表示．

三、常微分方程组求解

由若干个常微分方程联立起来的微分方程组，MATLAB 亦提供了求解办法，同样是用dsolve( )函数来求解．对于无初值条件的微分方程组，其通解的求解格式为

dsolve('微分方程 1,微分方程 2,…','自变量')

对于带初值条件的微分方程组，其特解的求解格式为：

dsolve('微分方程 1,微分方程 2,…','条件 1,条件 2,…','自变量')

其中，“微分方程 k”表示第 k 个微分方程，各微分方程间用逗号隔开，其表示方法与一个微分方程的表示方法相同，其它同前．例如求微分方程组

$$\begin{cases} f''+3g=\sin x, \\ g'+f'=\cos x, \\ f'(2)=0, \\ f(0)=0, \\ g(0)=0 \end{cases} \quad 和 \quad \begin{cases} f''+g'+h=\exp x, \\ f+g''+h'=\sin x, \\ f'+g+h''=\cos x, \\ f'(0)=f(0)=0, \\ g'(0)=g(0)=0, \\ h'(0)=h(0)=0 \end{cases}$$

的通解和特解，其命令如下（由于结果较长，故将结果略去）

```
syms x f g h
[f1,g1]=dsolve('D2f+3*g=sin(x),Df+Dg=cos(x)','x')
[f,g]=dsolve('D2f+3*g=sin(x),Df+Dg=cos(x)','Df(2)=0,f(0)=0,
      g(0)=0','x')
[f1,g1,h1]=dsolve('D2f+Dg+h=exp(x),f+D2g+Dh=sin(x),Df+g+D2h
      =cos(x)','x')
```

```
[f, g, h] = dsolve('D2f + Dg + h = exp(x), f + D2g + Dh = sin(x), Df + g + D2h
    = cos(x)','Df(0) = 0, f(0) = 0, Dg(0) = 0, g(0) = 0, Dh(0) = 0, h(0) = 0','x')
```

由其结果可知，对于常微分方程组来说，虽然可用 MATLAB 求出方程组的解，但随着方程个数的增加，解的表达式会十分复杂，而且给出的往往都是复数域上的解.

第四节　多元函数微积分

同一元函数一样，多元函数微积分也是微积分的重要组成部分.这一部分主要包括多元函数极限、偏导数、全导数、二重积分、三重积分等.由于多元函数的复杂性，在数学软件中没有直接求重极限和重积分的函数，因而必须经手工变为单次极限和单次积分后才能求出结果.

一、多元函数极限

一般来说，多元函数沿不同路径趋向于某一点时，可能会出现不同的结果.往往在有的时候沿某些特殊路径函数都有极限，但沿任意路径时函数却没有了极限，可见多元函数极限与路径是密切相关的.另一方面，函数趋向于某点的极限如果存在，则沿任何路径时的极限值均相同.基于这一点，这里我们仅对极限存在的函数，求沿坐标轴方向的极限，即将求多元函数极限问题，化成求多次单极限的问题.例如

```
syms x y z a b
limit(limit((x^2 + y^2)/(sin(x) + cos(y)), 0), pi)      ans = - pi^2
limit(limit(limit((x + y * z + exp(sin(x * z)))/(x + y/z), 1), 2), 3)
ans  = 21/5 + 3/5 * exp(sin(3))
limit(limit(x^2 + x * y + y^2, 2), 1)                    ans  = 7
limit(limit((1 - cos(x^2 + y^2))/((x^2 + y^2) * exp(x^2 * y^2)), 0), 0)
ans  = 0
```

但在 MATLAB 中却不能求出像

$$\lim_{\substack{x \to 0 \\ y \to 0}} \frac{2 - \sqrt{xy + 4}}{xy}$$

的极限，原因是第一次让 x 趋于零时同时消掉了 y，因而第二次就无法再对 y 求极限了.但在 MathCAD 中就不同了，MathCAD 可以正确求出上述二重极限.例如：