现代数值计算方法

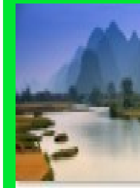
第六章 常微分方程的数值解法

Back
Close

第六章 常微分方程的数值解法

§6.5 一阶微分方程组和高阶微分方程

前面介绍了一阶常微分方程的各种数值方法, 这些方法对常微分方程组和高阶常微分方程同样适用. 为了避免书写上的复杂, 下面以二阶常微分方程和两个未知函数的方程组为例来叙述这些方法的计算公式, 其截断误差和推导过程与一阶的情形完全一样, 不再赘述, 只列出计算格式.



§6.5.1 一阶常微分方程组

考虑方程组

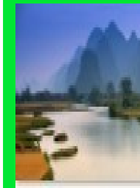
$$\begin{cases} y' = f(x, y, z), \quad y(x_0) = y_0, \\ z' = g(x, y, z), \quad z(x_0) = z_0. \end{cases} \tag{6.31}$$

(1) 欧拉格式

对 $n = 0, 1, 2, \cdots ,,$ 计算

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n, z_n), \quad y(x_0) = y_0, \\ z_{n+1} = z_n + hg(x_n, y_n, z_n), \quad z(x_0) = z_0. \end{cases} \tag{6.32}$$

(2) 改进欧拉格式



对 $n=0,1,2,\cdots,,$ 计算

$$
\begin{cases}
p_{n+1} = y_n + hf(x_n, y_n, z_n), \\
q_{n+1} = z_n + hg(x_n, y_n, z_n), \\
y_{n+1} = y_n + \dfrac{h}{2}\big[f(x_n, y_n, z_n) + f(x_{n+1}, p_{n+1}, q_{n+1})\big], \\
z_{n+1} = z_n + \dfrac{h}{2}\big[g(x_n, y_n, z_n) + g(x_{n+1}, p_{n+1}, q_{n+1})\big].
\end{cases}
\tag{6.33}
$$

其中 $y(x_0)=y_0,\ z(x_0)=z_0.$

(3) 经典四阶龙格-库塔格式

对 $n=0,1,2,\cdots,$ 计算

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\
z_{n+1} = z_n + \dfrac{h}{6}(L_1 + 2L_2 + 2L_3 + L_4),
\end{cases}
\tag{6.34}
$$

其中

$$
\begin{cases}
K_1 = f(x_n, y_n, z_n), \\
L_1 = g(x_n, y_n, z_n), \\
K_2 = f(x_n + \frac{h}{2}, y_n + \frac{k}{2}K_1, z_n + \frac{h}{2}L_1), \\
L_2 = g(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1, z_n + \frac{h}{2}L_1), \\
K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2, z_n + \frac{h}{2}L_2), \\
L_3 = g(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2, z_n + \frac{h}{2}L_2), \\
K_4 = f(x_n + h, y_n + hK_3, z_n + hL_3), \\
L_4 = g(x_n + h, y_n + hK_3, z_n + hL_3).
\end{cases}
$$

(4) Adams 外插格式

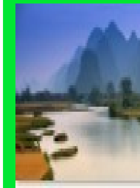
记 f_{n-k}, g_{n-k} 分别表示 $f(x_{n-k}, y_{n-k}, z_{n-k}), g(x_{n-k}, y_{n-k}, z_{n-k})$ $(k = 0, 1, 2, 3)$. 对 $n = 0, 1, 2, \cdots$, 计算

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{24}\left(55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}\right), \\ z_{n+1} = z_n + \dfrac{h}{24}\left(55 g_n - 59 g_{n-1} + 37 g_{n-2} - 9 g_{n-3}\right), \end{cases} \tag{6.35}$$

其中 $y(x_0) = y_0,\ z(x_0) = z_0$.

(5) Adams 预报-校正格式

记 f_{n-k}, g_{n-k} 分别表示 $f(x_{n-k}, y_{n-k}, z_{n-k}), g(x_{n-k}, y_{n-k}, z_{n-k})$ $(k =$



Back
Close

$0,1,2,3)$. 对 $n=0,1,2,\cdots$, 计算

$$\begin{cases} p_{n+1}=y_n+\dfrac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}), \\ q_{n+1}=z_n+\dfrac{h}{24}(55g_n-59g_{n-1}+37g_{n-2}-9g_{n-3}), \\ y_{n+1}=y_n+\dfrac{h}{24}(f_{n-2}-5f_{n-1}+19f_n+9f(x_{n+1},p_{n+1},q_{n+1})), \\ z_{n+1}=z_n+\dfrac{h}{24}(g_{n-2}-5g_{n-1}+19g_n+9g(x_{n+1},p_{n+1},q_{n+1})), \end{cases} \tag{6.36}$$

其中 $y(x_0)=y_0,\ z(x_0)=z_0$.

下面给出用经典四阶龙格-库塔格式解常微分方程组的 MATLAB 通用程序：

```
%marunge4s.m
function [x,y]=marunge4s(dyfun,xspan,y0,h)
```

```
%用途：4阶经典龙格库塔格式解方程组y'=f(x,y),y(x0)=y0
%格式：[x,y]=marunge4s(dyfun,xspan,y0,h)   dyfun为向
%       量函数f(x,y)，xspan为求解区间[x0,xn],y0为初值
%       向量，h为步长，x返回节点，y返回数值解向量
x=xspan(1):h:xspan(2);
y=zeros(length(y0),length(x));
y(:,1)=y0(:);
for n=1:(length(x)-1)
    k1=feval(dyfun,x(n),y(:,n));
    k2=feval(dyfun,x(n)+h/2,y(:,n)+h/2*k1);
    k3=feval(dyfun,x(n)+h/2,y(:,n)+h/2*k2);
    k4=feval(dyfun,x(n+1),y(:,n)+h*k3);
    y(:,n+1)=y(:,n)+h*(k1+2*k2+2*k3+k4)/6;
```

```
end
x=x'; y=y';
```

例 6.13 取 $h=0.02$, 利用程序 marunge4s.m 求刚性微分方程组

$$\begin{cases} y' = -0.01y - 99.99z, \quad y(0)=2, \\ z' = -100z, \quad z(0)=1, \end{cases}$$

的数值解, 其解析解为 $y=\mathrm{e}^{-0.01x}+\mathrm{e}^{-100x}$, $z=\mathrm{e}^{-100x}$.

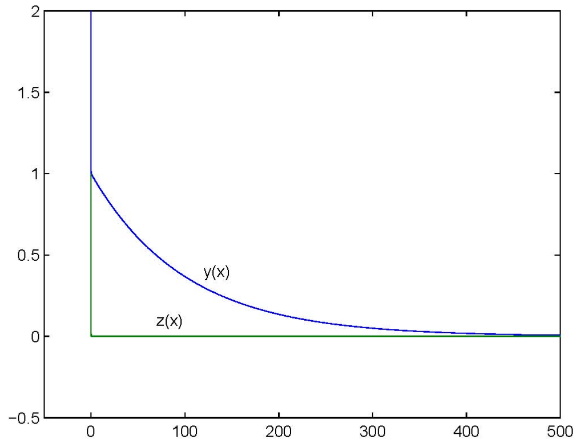
解 首先编写 M 函数 dyfun.m:

```
%dyfun.m
function f=dyfun(t,y)
f(1)=-0.01*y(1)-99.99*y(2);
f(2)=-100*y(2);
f=f(:);
```

再在 MATLAB 命令窗口执行：

```
>> [x,y]=marunge4s(@dyfun,[0 500],[2 1],0.02);
>> plot(x,y);
>> axis([-50 500 -0.5 2]);
>> text(120,0.4,'y(x)');
>> text(70,0.1,'z(x)');
```

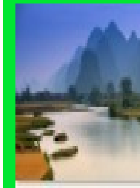
得到如下图所示的结果：



例 6.14 考虑下面的 Lornez 方程组

$$
\begin{cases}
\dfrac{\mathrm{d}x}{\mathrm{d}t} = -\sigma x + \sigma y, \\
\dfrac{\mathrm{d}y}{\mathrm{d}t} = \alpha x - y - xz, \\
\dfrac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.
\end{cases}
$$

参数 α, β, σ 适当的取值会使系统趋于混沌状态. 取 $\alpha = 30$, $\beta = 2.8$,



$\sigma = 12$, 利用经典四阶龙格-库塔法求其数值解, 并绘制 z 随 x 变化的曲线.

解 首先编写 M 函数 mafun.m:

```
%mafun.m
function ff=mafun(t,y)
b=2.8;r=30;sigma=12;
ff(1)=-sigma*y(1)+sigma*y(2);
ff(2)=r*y(1)-y(2)-y(1)*y(3);
ff(3)=y(1)*y(2)-b*y(3);
ff=ff(:);
```

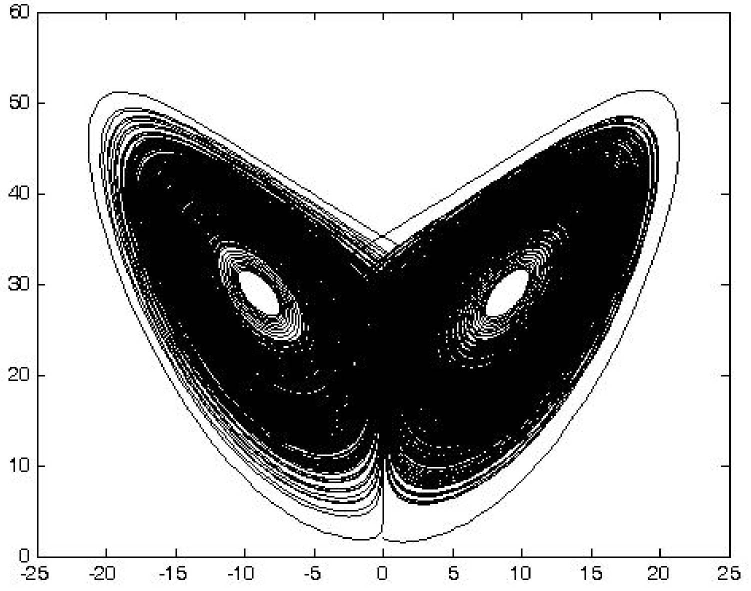
再在 MATLAB 命令窗口执行:

```
>>  [t,y]=marunge4s(@mafun,[0 500],[0 1 2],0.005);
```

Back
Close

```
>>  plot(y(:,1), y(:,3),'r');
```

得到如下图所示的结果：



§6.5.2 高阶常微分方程

对于高阶常微分方程, 它总可以化成方程组的形式. 例如, 二阶方程

$$\begin{cases} y'' = g(x, y, y'), \\ y(x_0) = y_0, \quad y'(x_0) = y'_0, \end{cases} \tag{6.37}$$

总可以化为一阶方程组

$$\begin{cases} y' = z, \\ z' = g(x, y, z), \\ y(x_0) = y_0, \quad z(x_0) = y'_0 = z_0. \end{cases} \tag{6.38}$$

所以没有必要再对高阶方程给出计算公式. 但应注意到, 把高阶方程化为方程组时, 其函数取特定的形式. 因此, 这时的计算公式可以化简. 例如, 对改进欧拉格式, 因 $f(x, y, z) = z$, 故公式可表示为

Back
Close

对 $n=0,1,2,\cdots$, 计算

$$\begin{cases} p_{n+1}=y_n+hz_n, \\ q_{n+1}=z_n+hg(x_n,y_n,z_n), \\ y_{n+1}=y_n+\dfrac{h}{2}\left(z_n+q_{n+1}\right) \\ z_{n+1}=z_n+\dfrac{h}{2}\left[g(x_n,y_n,z_n)+g(x_{n+1},p_{n+1},q_{n+1})\right], \end{cases} \tag{6.39}$$

其中 $y(x_0)=y_0,\ z(x_0)=y_0'=z_0$.

例 6.15 取 $h=0.1$, 利用程序 marunge4s.m 求 2 阶方程

$$\begin{cases} y''=2y^3, \quad 1\le x\le 1.5, \\ y(1)=y'(1)=-1 \end{cases}$$

的数值解, 其解析解为

$$y=\frac{1}{x-2}.$$

解　首先将 2 阶方程写成一解方程组的形式

$$\begin{cases} y' = z, \quad y(1) = -1, \\ z' = 2y^3, \quad z(1) = -1. \end{cases}$$

再编写 M 函数 dyfun1.m:

```
%function f=dyfun1(t,y)
f(1)=y(2);
f(2)=2*y(1)^3;
f=f(:);
```

然后再在 MATLAB 命令窗口执行：

```
>> [x,y]=marunge4s(@dyfun1,[1 1.5],[-1 -1],0.1);
>> y1=1./(x-2);  %求精确解
>> [x,y(:,1),y1]
```

Back
Close

```
ans =

      1.0000   -1.0000   -1.0000

      1.1000   -1.1111   -1.1111

      1.2000   -1.2500   -1.2500

      1.3000   -1.4285   -1.4286

      1.4000   -1.6666   -1.6667

      1.5000   -1.9998   -2.0000
```

上面的显示结果, 第 1 列是节点, 第 2 列是数值解, 第 3 列是精确解.

作业：P145: 6.16; P146: 6.18.