第3章

基 础 实 验

3.1 基本数值积分方法仿真

1. 实验目的

(1) 学会编写 MATLAB 的 M 程序文件。
(2) 掌握基本数值积分方法(欧拉法、梯形法、四阶龙格-库塔法)原理和仿真方法。
(3) 学会使用 MATLAB 的 ode45 函数。

2. 实验原理

1) 欧拉法
假设一阶微分方程为

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}t}=f(t,y)\\ y(t_0)=y_0\end{cases} \tag{3-1}$$

针对该方程,在区间(t_k,t_{k+1})上求积分,有

$$y(t_{k+1})=y(t_k)+\int_{t_k}^{t_{k+1}}f(t,y)\mathrm{d}t \tag{3-2}$$

根据积分的几何意义可知,$\int_{t_k}^{t_{k+1}}f(t,y)\mathrm{d}t$是曲线$f(t,y)$在区间$(t_k,t_{k+1})$上的面积,当区间$(t_k,t_{k+1})$足够小时,可用矩形面积近似代替(如图 3.1 所示),即

$$\int_{t_k}^{t_{k+1}}f(t,y)\mathrm{d}t\approx(t_{k+1}-t_k)f(t_k,y_k)=hf(t_k,y_k)$$

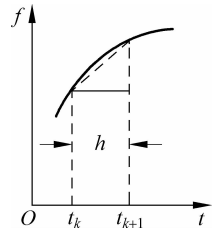

图 3.1 积分图

其中 $h=t_{k+1}-t_k$ 称为步长。

因此式(3-2)可近似为

$$y(t_{k+1}) \approx y(t_k)+hf(t_k,y_k)$$

进而将该式写成递推差分格式为

$$y_{k+1} = y_k + hf(t_k,y_k) \tag{3-3}$$

2) 梯形法

针对式(3-2),可以改进欧拉法的方法,即使用梯形面积代替矩形面积,得到

$$y(t_{k+1}) \approx y(t_k)+\frac{h}{2}[f(t_k,y_k)+f(t_{k+1},y_{k+1})]$$

进而将该式写成递推差分格式为

$$y_{k+1} = y_k + \frac{h}{2}[f(t_k,y_k)+f(t_{k+1},y_{k+1})] \tag{3-4}$$

从式(3-4)可以看出,在计算等式左侧的 y_{k+1}时,等式右侧却已经存在 y_{k+1}。一般利用欧拉法先进行一次计算求出 y_{k+1}的预估值,然后再将这个预估值代入到式(3-4)中计算 y_{k+1}的校正值,从而构成预估-校正公式为

$$y_{k+1}^0 = y_k + hf(t_k,y_k) \tag{3-5}$$

$$y_{k+1} = y_k + \frac{h}{2}[f(t_k,y_k)+f(t_{k+1},y_{k+1}^0)] \tag{3-6}$$

3) 龙格-库塔法

针对一阶微分方程式(3-1),假设其精确解是充分光滑的,故可将其解 $y(t)$在 t_k 附近用泰勒级数展开,只保留 h^2 项,则泰勒级数展开式为

$$y(t_{k+1}) = y(t_k+h) = y(t_k)+h\dot{y}(t_k)+\frac{1}{2!}h^2\ddot{y}(t_k) \tag{3-7}$$

已知

$$y(t_{k+1}) = y_{k+1}, y(t_k) = y_k, \dot{y}(t_k) = f(t_k,y_k)$$

则有

$$\ddot{y}(t_k) = \frac{\mathrm{d}^2y}{\mathrm{d}t} = \frac{\mathrm{d}f(t_k,y_k)}{\mathrm{d}t} = \left[\frac{\partial f}{\partial t}+f(t_k,y_k)\frac{\partial f}{\partial y}\right]_{\substack{t=t_k\\y=y_k}}$$

则式(3-7)得

$$y_{k+1} = y_k + hf(t_k,y_k)+\frac{h^2}{2}\left[\frac{\partial f}{\partial t}+f(t_k,y_k)\frac{\partial f}{\partial y}\right]_{\substack{t=t_k\\y=y_k}} \tag{3-8}$$

将式(3-8)写成如下形式

$$y_{k+1} = y_k + h(a_1k_1+a_2k_2) \tag{3-9}$$

$$k_1 = f(t_k,y_k) \tag{3-10}$$

$$k_2 = f(t_k+b_1h, y_k+b_2k_1h) \tag{3-11}$$

将式(3-11)右侧的函数在 t_k 附近用泰勒级数展开，只保留 h 项，则得

$$k_2 = f(t_k, y_k) + h\left[b_1 \frac{\partial f}{\partial t} + b_2 k_1 \frac{\partial f}{\partial y}\right]_{\substack{t=t_k \\ y=y_k}} \tag{3-12}$$

将 k_1 式(3-10)和 k_2 式(3-12)代入到式(3-9)中，则有

$$y_{k+1} = y_k + a_1 h f(t_k, y_k) + a_2 h f(t_k, y_k) + a_2 h^2 \left[b_1 \frac{\partial f}{\partial t} + b_2 k_1 \frac{\partial f}{\partial y}\right]_{\substack{t=t_k \\ y=y_k}} \tag{3-13}$$

将式(3-13)和式(3-8)比较，则有

$$\begin{cases} a_1 + a_2 = 1 \\ a_2 b_1 = \dfrac{1}{2} \\ a_2 b_2 = \dfrac{1}{2} \end{cases} \tag{3-14}$$

方程组(3-14)中，有 4 个未知数，解不能唯一，因此令 $a_1 = a_2$，则有 $a_1 = a_2 = \frac{1}{2}$，$b_1 = b_2 = 1$，然后将这几个参数代入式(3-9)，得

$$\begin{cases} y_{k+1} = y_k + \dfrac{h}{2}(k_1 + k_2) \\ k_1 = f(t_k, y_k) \\ k_2 = f(t_k + h, y_k + k_1 h) \end{cases} \tag{3-15}$$

由于式(3-15)只取泰勒级数展开式的前三项，只保留到 h^2 项，高阶项省略，故称为二阶龙格-库塔法。二阶龙格-库塔法的计算精度不高，实际上与梯形法等价。在实际工程应用中，经常采用的是精度更高的四二阶龙格-库塔法，其递推公式为

$$\begin{cases} y_{k+1} = y_k + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(t_k, y_k) \\ k_2 = f\left(t_k + \dfrac{h}{2}, y_k + \dfrac{h}{2}k_1\right) \\ k_3 = f\left(t_k + \dfrac{h}{2}, y_k + \dfrac{h}{2}k_2\right) \\ k_4 = f(t_k + h, y_k + hk_3) \end{cases} \tag{3-16}$$

4) ode45 函数

ode45 函数，采用四阶-五阶 Runge-Kutta 算法，用四阶方法提供候选解，用五阶方法控制误差，是一种自适应步长(变步长)的求常微分方程数值解法。其常用格式为

```
[t,Y] = ode45(odefun,tspan,y0)
```

其中，odefun 为定义的常微分方程函数名，tspan[t0,Final]为起止时间向量，y0 为初始状

态向量。

3. 实验内容

求解微分方程 $dy/dt=-30y, y(0)=1/3$。采用欧拉法、梯形法、四阶龙格-库塔法等基本数值积分方法进行仿真，并运用 MATLAB 所提供的 ode45 函数法、直接求解法求解。分别在仿真步长为 0.1 和 0.01 条件下，记录所有仿真方法的 $y(1.5)$值，以及 $0\leqslant t\leqslant 1.5$ 范围内的仿真图形，对比并分析结果。

提示：微分方程 $dy/dt=-30y$，可以按照状态方程形式，得到参数 $A=-30, B=0, C=1, D=0$。

4. 实验报告要求

(1) 描述实验原理。
(2) 完整的源代码。
(3) 记录所有仿真方法求 $y(1.5)$的值。
(4) 通过抓图方式，获取所有仿真方法在时间 $0\leqslant t\leqslant 1.5$ 范围内的仿真图形。
(5) 对比实验数据并分析，写出分析结论。

5. 参考源程序

```
%求解 Dy/Dt = -30*y 并调用函数文件 rr.m
function f = main
syms y t;                              %符号定义或者: y = sym('y') 或者: y = 'y'
y = dsolve('Dy = -30*y','y(0) = 1/3','t');
                                       %解单个微分方程或微分方正组求: dy/dt = --30y
y = simplify(y)                        %化简得 y = y = 1/3*exp(-30*t)
ezplot(y)                              %绘制符号形式 y(t)的图形

disp('精确解: ')
t =1.5;                                %求 t = 1.5 时刻的 y 值
y = 1/3*exp(-30*t)                     %y = 9.5417e-021
h = 0.01;s = 1.5;t1 = 0;y1 = 1/3*exp(-30*t1);     %初始参数
m = s/h;                               %迭代次数
for i = 1:m
    t1 = [t1,t1(i) + h];
    y1 = [y1,1/3*exp(-30*t1(i+1))];
end
figure(2)                              %第 2 个图形
plot(t1,y1)                            %精确解画图
```

```
xlabel('t(s)')                                  % 图形 x 轴坐标标示
ylabel('y')                                     % 图形 y 轴坐标标示
title('dy/dt = -30y,y(0) = 1/3,y(1.5)?')        % 图形标题
legend('y = 1/3 * exp( -30 * t)')               % 图形曲线说明

figure(3)                                       % 第 3 个图形
disp('求 s = 1.5 秒时刻的解: ');s = 1.5;
disp('步长取: ');h = 0.1
u = 0;t1 = 0;x1 = 1/3;y1 = x1;                  % 初始参数
a = -30;b = 0;c = 1;                            % 初始参数
m = s/h;                                        % 迭代次数

disp('欧拉法: ')
for i = 1:m
    x1 = x1 + h * (a * x1 + b * u);
    y1 = [y1,c * x1];
    t1 = [t1,t1(i) + h];
end
subplot(2,2,1)                                  % 设置 2 行 2 列子图中第 1 个子图
plot(t1,y1)                                     % 欧拉法画图
xlabel('t(s)')                                  % 图形 x 轴坐标标示
ylabel('y')                                     % 图形 y 轴坐标标示
title('欧拉法')                                 % 图形标题
legend('y')                                     % 图形曲线说明
y1(m + 1)                                       % 欧拉法 t = 1.5 时刻的 y 值
disp('梯形法: ')
u = 0;t1 = 0;x1 = 1/3;y1 = x1;                  % 初始参数
for i = 1:m
    x11 = x1 + h * (a * x1 + b * u);
    x1 = x1 + 0.5 * h * ((a * x1 + b * u) + (a * x11 + b * u));
    y1 = [y1,c * x1];
    t1 = [t1,t1(i) + h];
end
subplot(2,2,2)                                  % 设置 2 行 2 列子图中第 2 个子图
plot(t1,y1)                                     % 梯形法画图
xlabel('t(s)')                                  % 图形 x 轴坐标标示
ylabel('y')                                     % 图形 y 轴坐标标示
title('梯形法')                                 % 图形标题
legend('y')                                     % 图形曲线说明
y1(m + 1)                                       % 梯形法 t = 1.5 时刻的 y 值

disp('四阶龙格 - 库塔法: ')
u = 0;t1 = 0;x1 = 1/3;y1 = x1;                  % 初始参数
```

```
for i = 1:m
    k1 = a * x1 + b * u;
    k2 = a * (x1 + h * k1/2) + b * u;
    k3 = a * (x1 + h * k2/2) + b * u;
    k4 = a * (x1 + h * k3) + b * u;
    x1 = x1 + h * (k1 + 2 * k2 + 2 * k3 + k4)/6;
    y1 = [y1,c * x1];
    t1 = [t1,t1(i) + h];
end
subplot(2,2,3)                          % 设置 2 行 2 列子图中第 3 个子图
plot(t1,y1)                             % 四阶龙格 - 库塔法画图
xlabel('t(s)')                          % 图形 x 轴坐标标示
ylabel('y')                             % 图形 y 轴坐标标示
title('四阶龙格 - 库塔法')              % 图形标题
legend('y')                             % 图形曲线说明
y1(m + 1)                               % 四阶龙格 - 库塔法 t = 1.5 时刻的 y 值

clear
disp('ode45 法: ')
x1 = 1/3;                               % 初始参数
 % [tt,yy] = ode45('rr',[0 1.5],x1); % 调用 rr.m 文件
[tt,yy] = ode45(@rr,[0 1.5],x1);
                        % 用符号@可以调用本程序内部定义的函数,否则只能调用外部文件 rr.m
subplot(2,2,4)                          % 设置 2 行 2 列子图中第 4 个子图
plot(tt,yy)                             % ode45 法画图
xlabel('t(s)')                          % 图形 x 轴坐标标示
ylabel('y')                             % 图形 y 轴坐标标示
title('ode45 法')                       % 图形标题
legend('y')                             % 图形曲线说明
fprintf('%e',yy(101))                   % ode45 法 t = 1.5 时刻的 y 值

function xd = rr(t,y)                   % 定义被 ode45 调用的函数
u = 0;
xd = -30 * y;
```

6. 思考问题

(1) 仿真步长的作用是什么？能否采用变步长？

(2) 请自行选择一个闭环传递函数作为数学模型进行仿真。

3.2 传递函数与状态方程模型仿真

1. 实验目的

(1) 掌握数学模型是传递函数的仿真方法。
(2) 掌握数学模型是状态方程的仿真方法。
(3) 学会使用 step 函数。

2. 实验原理

(1) 数值积分仿真方法见 3.1 节实验原理部分。
(2) 数学模型建立、数学模型转换、反馈函数等内容见 2.4.2 节。
(3) MATLAB 内部库函数 step 见 2.4.2 节。

3. 实验内容

已知系统数学模型是传递函数形式，如 $G(s)=\dfrac{2(s+2)}{s(s+3)(s^2+2s+2)}$，求其单位负反馈闭环系统的阶跃响应。要求使用数学模型是传递函数和状态方程的形式进行仿真。

4. 实验报告要求

(1) 描述实验原理。
(2) 完整的源代码。
(3) 记录不同方法所得的闭环传递函数模型。
(4) 通过抓图方式，获取所有仿真方法的仿真图形。
(5) 对比实验数据并分析，写出分析结论。

5. 参考源程序

```
clear all                                    % 清除所有变量
num = 2 * [1 2];                             % 传递函数分子
den = conv(conv([1 0],[1 3]),[1 2 2]);       % 传递函数分母

 % 方法一：使用传递函数进行负反馈，并用闭环传递函数仿真
sys = tf(num,den);                           % 传递函数模型
```

```
sys1 = feedback(sys,1, - 1);          % 传递函数负反馈
figure                                % 打开新窗口
step(sys1,[0:0.01:35]);               % 求系统的单位阶跃响应
grid                                  % 网格开

% 方法二: 使用传递函数进行负反馈,将闭环传递函数转换为状态方程
sys = tf(num,den);                    % 传递函数模型
sys1 = feedback(sys,1, - 1);          % 传递函数负反馈
[num1,den1] = tfdata(sys1,'v')        % 通过参数'v'得到传递函数的参数(分子,分母值)
[a,b,c,d] = tf2ss(num1,den1)          % 通过传递函数的参数(分子,分母值)得到状态方程参数
sys = ss(a,b,c,d);                    % 状态方程模型
figure                                % 打开新窗口
step(sys,[0:0.01:35]); grid           % 求系统的单位阶跃响应

% 方法三: 使用传递函数转换为状态方程,然后负反馈
[a,b,c,d] = tf2ss(num,den)            % 状态方程参数
sys = ss(a,b,c,d);                    % 状态方程模型
sys1 = feedback(sys,1);               % 状态方程负反馈
[a,b,c,d] = ssdata(sys1)              % 负反馈后状态方程参数
sys = ss(a,b,c,d);                    % 状态方程模型
figure                                % 打开新窗口
step(sys,[0:0.01:35]);grid            % 求系统的单位阶跃响应

% 方法四: 使用传递函数转换为状态方程,采用公式法进行负反馈
[a,b,c,d] = tf2ss(num,den)            % 状态方程参数
a = a - b * c                         % 公式法负反馈
h = 0.01;tf = 30;u = 1;               % 仿真步长、总时间、输入响应
y = 0;t = 0;x = [0 0 0 0]';           % 初始值
m = tf/h;                             % 迭代次数
for i = 1:m                           % 四阶龙格 - 库塔法
  k1 = a * x + b * u;
  k2 = a * ( x + h * k1/2) + b * u;
  k3 = a * ( x + h * k2/2) + b * u;
  k4 = a * ( x + h * k3) + b * u;
  x = x + h * (k1 + 2 * k2 + 2 * k3 + k4)/6;
  y = [y,c * x];                      % 输出
  t = [t,t(i) + h];                   % 仿真时间
end
figure                                % 打开新窗口
plot(t,y);grid                        % 画图
xlabel('时间 Time(s)');ylabel('输出 y');   % 横坐标、纵坐标标示
```

6. 思考问题

(1) 采用零极点增益模型方式进行仿真。
(2) 采用 Simulink 工具建立模型并仿真。

3.3 面向控制系统结构图的仿真

1. 实验目的

(1) 学会使用 Simulink 图形仿真工具。
(2) 学会使用 Connect 函数。
(3) 掌握数学模型串联、反馈等方法。

2. 实验原理

(1) Simulink 图形仿真工具见 2.5 节。
(2) 数学模型串联、并联函数等内容见 2.4.2 节。

3. 实验内容

针对图 3.2 给出的一个控制系统结构图模型进行仿真。分析图 3.2,可以确定为三个典型模块,模块 1 为积分环节$\frac{2}{s}$,模块 2 为惯性环节$\frac{1}{s+1}$,模块 3 为增益环节 2。

要求使用三种方法进行仿真。

1) 使用 Simulink 工具仿真

使用 Simulink 图形工具绘制结构图,可以直接参考图 3.2。

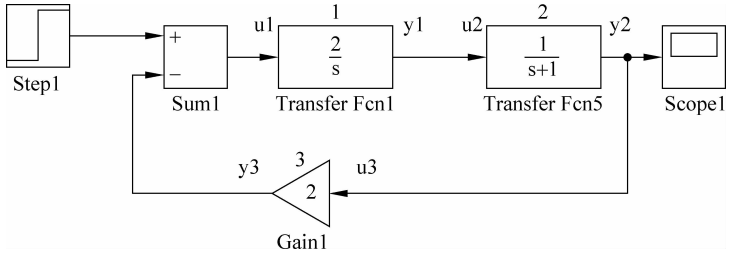

图 3.2 结构图模型

2）使用 Connect 函数仿真

使用 Connect 连接函数获取系统状态方程参数，然后采用四阶龙格-库塔法仿真。

3）使用数学模型串联、反馈方法仿真

将若干模块的数学模型，以串联、并联和反馈方式组合成为一个整体，然后采用四阶龙格-库塔法仿真。

4. 实验报告要求

（1）描述实验原理。

（2）完整的源代码。

（3）记录不同方法所得的闭环传递函数模型。

（4）通过抓图方式，获取所有仿真方法的仿真图形。

（5）对比实验数据并分析，写出分析结论。

5. 参考源程序

1）Connect 函数仿真

```
clear all                          %清除所有变量
nblocks = 3;                       %固定格式:设定环节数量
n1 = 2;d1 = [1 0];                 %固定格式:环节 1 的 n1 和 d1 分别表示传递函数的分子和分母
n2 = 1;d2 = [1 1];                 %固定格式:环节 2 的 n2 和 d2 分别表示传递函数的分子和分母
n3 = -2;d3 = 1;                    %固定格式:环节 3 的 n3 和 d3 分别表示传递函数的分子和分母
blkbuild;                          %固定格式:创建状态参数
Q = [1 3 ;          %固定格式:第 1 列为环节编号; 第 2,3,...,n 列为进入该环节的所有环节编号
  2 1 ;
  3 2];
inputs = 1;                        %输入量进入的环节编号
outputs = 2;                       %输出量对应的环节编号,如果是多输出,则为向量
[A,B,C,D] = connect(a,b,c,d,Q,inputs,outputs)          %构造数学模型并获得状态方程参数
a = A;b = B;c = C;d = D;
h = 0.01;tf = 29;y = 0;t = 0;u = 1;x = [0 0]';                %初始参数
m = tf/h;                          %迭代次数
for i = 1:m                        %四阶龙格-库塔法
  k1 = a * x + b * u;
  k2 = a * ( x + h * k1/2) + b * u;
  k3 = a * ( x + h * k2/2) + b * u;
  k4 = a * ( x + h * k3) + b * u;
  x = x + h*(k1 + 2*k2 + 2*k3 + k4)/6;
```