

```

% exp7_2.m --- 二阶 R-K 法

function try_myode2
% 下面的例题就 P187 实验课题(三)(3) 书上有精确解可对比

a = 0; b = 1; n = 10;
Y0 = [1,1,-1]';

[x,Y] = myode2(@F,a,b,n,Y0);

% 画第二个函数精确解,并与数值解比较,其它同样可画
xx = linspace(0,1,20); yy = xx.^3 + 1;
plot(x,Y(:,2),'+',xx,yy)
legend('数值解','精确解')

% ----- 二阶 Runge-Kutta 法 -----
function [x,Y] = myode2(F,a,b,n,Y0)
% [x,Y] = myode2(F,a,b,n,Y0) -- 改进的 Euler 法,见 P164(7-15) 式
% 输入   F --- 方程(组)的函数(返回值必须是列向量或标量)
%       a,b --- 积分区间两个端点
%       n --- 区间等分数
%       Y0 --- 初值,必须是列向量或标量
% 输出   x --- [a,b]等分节点(列向量)
%       Y --- 第 k 列为第 k 个函数的数值解

h = (b-a)/n;
x = linspace(a,b,n+1)';
m = length(Y0);
Y = zeros(n+1,m);

Y(1,:) = Y0';
y = Y0;
for j=1:n
    t = x(j);
    k1 = feval(F, t, y );
    k2 = feval(F, t+h, y+h*k1);
    y = y + h*( k1+k2)/2;
    Y(j+1,:) = y';
end

% -----
function Z = F(x,Y)
Z=[ Y(2)-Y(3)+x
    3*x^2

```

```
Y(2)+exp(-x)];  
% -----  
  
% ***** 你的实验 *****  
% ★【实验一】  
% 把此程序,改为 4 阶标准 R-K 法,也做此例子  
% 你可以把此程序另存为一个新名子,直接修改
```