```
% exp2_1.m --- 学习非线性方程 f(x) = 0 数值求根命令 fzero
% [调用]          X = fzero(FUN,X0), FUN 是函数,求 X0 附近的根
%          或    X = fzero(FUN,[a b]), 求 [a,b] 之间的根,这时 FUN(a) 与 FUN(b) 符号必须相反
% [注]   该命令集二分法, 割线法, 逆二次内插法于一体,
%          该命令只适用于奇重根,即在根的附近左右函数值反号

function study_fzero

% example1:求 sin(x)在 3 附近的根
X1 = fzero(@sin,3)

% example2: 求 f1 在[0,1]之间的根, 其中 f1 是用内联方式定义的函数
f1 = inline('cos(x)-x');
X2 = fzero(f1,[0 1])

% example3: 求 f2 = x-exp(-x) = 0 的根
%          首先通过作图估计根的大致位置(近似值)

f2 = inline('x-exp(-x)');   % 定义函数
x = linspace(0,1,101);      % 采样: 把 [0,1] 100 等分,x 为 101 维的行向量
y = f2(x);                  % 计算 f2 的函数值,其维数同 x
plot(x,y,'r')               % 作图(就是把各个点连起来),颜色红色
grid on                     % 显示坐标刻度网格
title('f(x) = x - e^{-x} = 0 的根')          % 加标题
text(0.3,0,'根的大致位置\rightarrow')          % 加文字说明
                            % 通过作图知道在[0.5,0.6]之间有一个根

%          再调用 fzero 命令求更精确的近似值,参见 P23 例 6
X3 = fzero(f2,[0.5 0.6])




% ********   你的实验    ********

% ★【实验一】 割线法( 见 P24 (2-13)式 )
% [方法]   设 a,b 为迭代初值,求两点(a,f(a)) 与 (b,f(b)) 的连线(割线)与 x 轴的交点记为 c
%          再把迭代初值换成 b,c,重复计算.
% [要求]   把下面程序复制为新的 M-文件,去掉开头的 %
%          再把 '?' 部分改写正确就是一个完整的程序,找前面一个例子试算

% function mysecant
% f = inline('?');
```

```
% a = ?;    b = ?;
% delta = ?; epsilon = ?; max1 = ?;
% [c,err,iter,yc] = secant(f,a,b,delta,epsilon,max1)
%
% % ------------------------------------------------------------
% function [c,err,iter,yc] = secant(f,a,b,delta,epsilon,max1)
% % [c,err,iter,yc] = secant(f,a,b,delta,epsilon,max1)
% %  输入:   f  连续函数
% %              a,b  迭代初值
% %              delta,epsilon  容差
% %              max1  最大迭代次数
% %  输出:   c  近似根
% %              err  误差
% %              iter  迭代次数
% %              yc = f(c)
%
% for k = 1:max1
%       ya = feval(f,a);                  % ya = f(a)
%       yb = feval(f,b);
%       c = ?;                                % 割线与 x 轴交点的横坐标
%
%       err = abs(c-b);                   % 相邻两次迭代的误差
%       relerr = err/(abs(c)+eps);   % 相对误差,eps 是 matlab 常数(机器精度)约为 1e-16
%                                             % 为什么分母要加上一个小常数?
%
%       yc = feval(f,c);
%
%       if (err<delta) | (relerr<delta) | (abs(yc)<epsilon)    % '|'是'或'
%             break
%       end
%
%       a = ?;
%       b = ?;
% end
% iter = ?;
% % ------------------------------------


% 【实验二】   绘制的隐函数的图像
% [方法]   众所周知,隐函数一般是不能用显式方式表示的, 故确定隐函数的大致图像是非常重要的.
%             对于方程 F(x,y) = 0 如果固定 x 就是一个关于 y 的非线性方程,我们可以通过求根的方法求出 y
%             因此只要对 x 离散化 x(k),k = 1,2,...,再求得 y(k) ,把点( (x(k),y(k)) )连起来
```

```
%              就能得到由方程 F(x,y) = 0 所确定的隐函数 y = f(x) 的大致图像
% [要求]   绘制由下面方程所确定的隐函数 y = f(x) 的图像
%                                  y^3
%              F(x,y) = ———————————— + x^2 - 4x = 0 ,        -5 <= x <= 5
%                              2 + 0.1sin(xy)
%              这里把 [-5,5] 用 linspace 命令 100 等分
%              第一次初值用 y0 = -4.6, 以后用 y(k) 作为下一次求 y(k+1) 的迭代初值(想一想有无道理?)
%              按照上面要求,请你把下面程序中的 '?' 部分填写正确了就是一个完整的程序
% [技巧]   由于 x(k) 变化, 每次函数 F[x(k),y] 也在变,你可以用一个全局变量解决此问题
% [注]      参考 ezplot 作图命令.


% %  隐函数作图
% function implicit_function
% global p                          %  定义全局变量
% n = ?;
% x = linspace(?);
% y = zeros(1,n);                %  定义矩阵,初值是零,这是最常用的定义矩阵的方法
% y0 = -4.6;                       %  第一次迭代初值
% for k = 1:n
%        p = ?;
%        y(k) = fzero(@fun,y0);
%        y0 = ?;
% end
% plot(?)                           %  作图
% title(?)                          %  加个标题
% %------------------------------
% function z = fun(y)           %  定义函数,这是最常用的定义函数的方式
% global p
% x = p;
% z = y^3/(2+0.1*sin(x*y))+x^2-4*x;
```