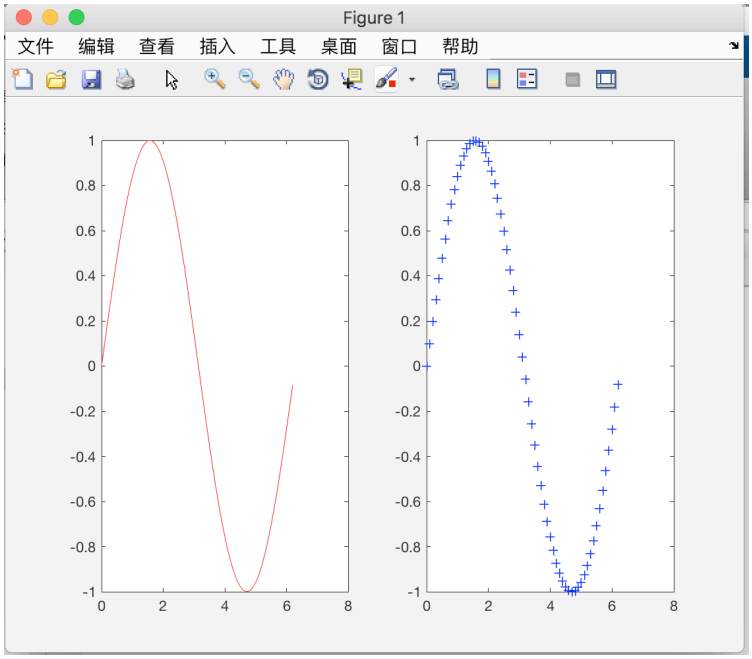
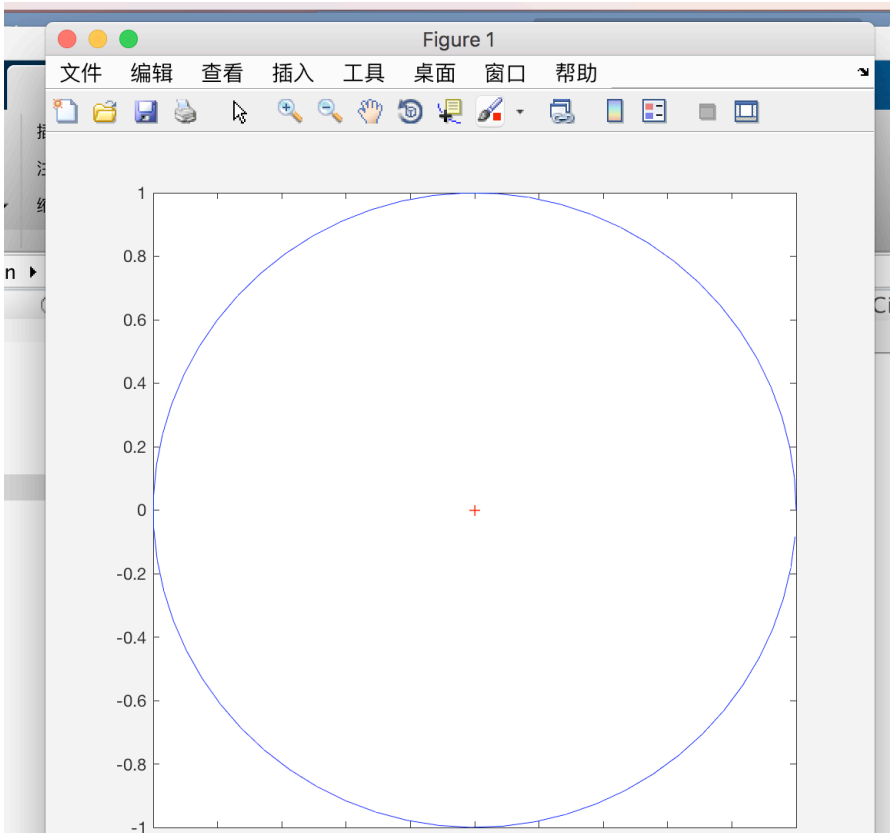
实验一　中点 Bresenham 算法

学号：1525161007　姓名：秦源　班级：软件工程一班

2、新建一个 M 文件，在其中输入下列语句，观察显示结果
(实验截图)



3、试在 Matlab 中绘制一个单位圆，圆心用红色”+”标出。
(实验截图)



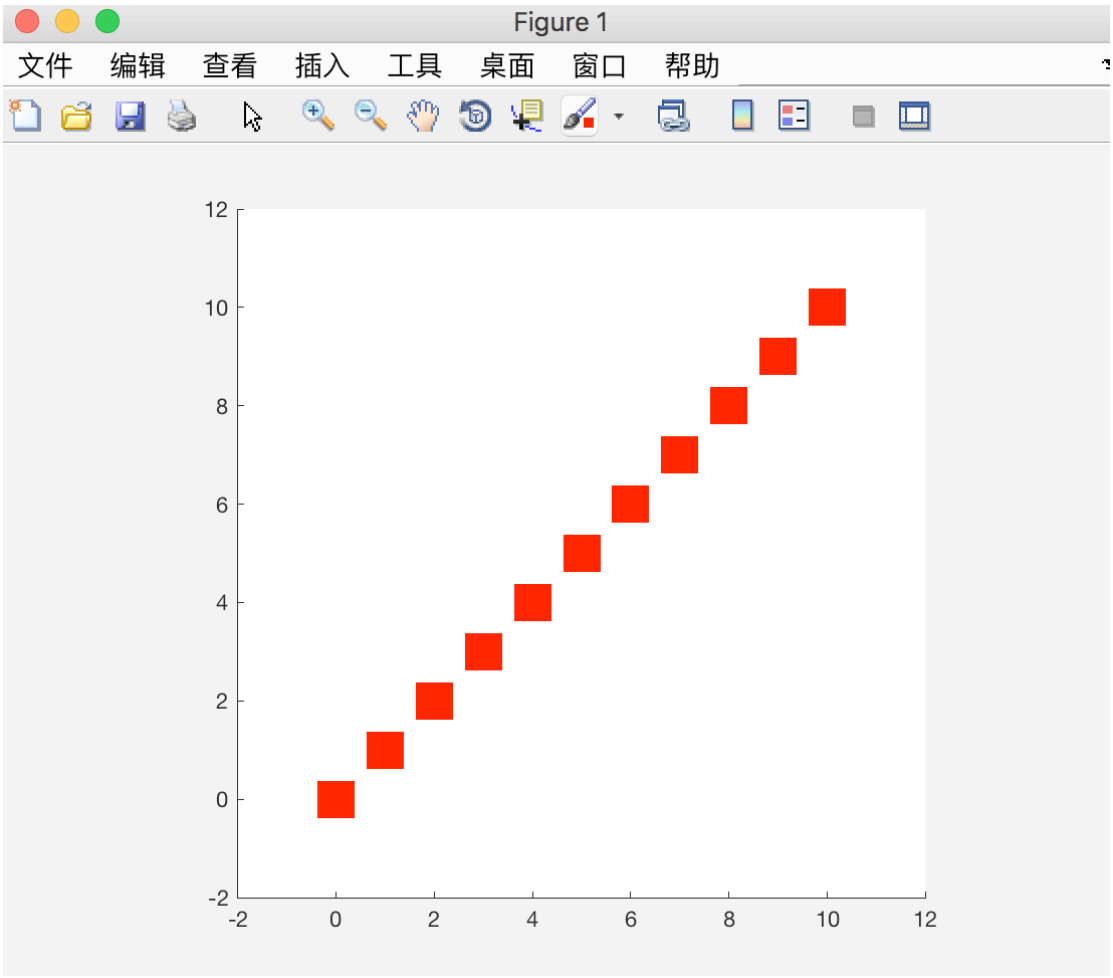
4. 中点 Bresenham 画直线的算法

(1) 观察下列程序，将空格处的注释语句填写完整

```
%中点 Bresenham 画直线的算法
% P,Q 为直线段端点，pixs_line 为计算出的像素点
function pixs_line = bresenham_line(P,Q)

%  规格化直线 PQ，使 PQ 的斜率:0<=k<=1
%  当直线 PQ 的斜率小于 0 时,将 Flag(1)置 1,并将直线沿 y 轴翻转
%  当直线 PQ 的斜率绝对值大于 1 时,将 Flag(2)置 1,并将直线沿 y=x 翻转
%  当直线 PQ 的 P 点在 Q 点的右上方时,将 Flag(3)置 1,并将 P 点 Q 点互换
Flag = [0 0 0];
if (Q(2)-P(2))*(Q(1)-P(1)) < 0
      Flag(1) = 1;
      P(2) = -P(2); Q(2) = -Q(2);
end
if abs(Q(2)-P(2)) > abs(Q(1)-P(1))
      Flag(2) = 1;
      P = [P(2) P(1)];    Q = [Q(2),Q(1)];
end
if P(1) > Q(1)
      Flag(3) = 1;
      t = P; P = Q; Q = t;
end

%  初始化变量
n = abs(Q(1)-P(1)) + 1; %  横轴像素点个数
pixs_line = zeros(n,2); %  初始化坐标数组 size:n*2
pixs_line(1,1:2) = [P(1) P(2)]; %  数组第一个点为 P 点

%  计算初始值 dx,dy,d=dx-2dy
dx = Q(1) - P(1);    dy = Q(2) - P(2);
d = dx - 2*dy;   %  由于只要 di 的符号，可以用 2di*dx 代替 di 来摆脱小数
for i = 2:n
      pixs_line(i,1) = P(1) + i - 1;
      %  当 d<0，取右上方像素点,d 的增量为 1-k
      if d < 0
            pixs_line(i,2) = pixs_line(i-1,2) + 1;
            d = d + 2*dx - 2*dy;
      else
            %  当 d>=0，取正右方像素点,d 的增量为-k
            pixs_line(i,2) = pixs_line(i-1,2);
            d = d - 2*dy;
      end
end

%  还原原始直线 PQ
if Flag(3) == 1
      pixs_line = flipud(pixs_line); %  矩阵上下翻转,相当于交换 P 点和 Q 点
end
if Flag(2) == 1
      pixs_line = fliplr(pixs_line); %  矩阵左右翻转,相当于 PQ 沿 y=x 翻转
end
if Flag(1) == 1;
      %  矩阵第二列*-1,相当于 PQ 沿 y 轴翻转
      pixs_line = [pixs_line(:,1) -1*pixs_line(:,2)];
end
```

(3) 使用下列函数逐点绘制生成的直线段

(实验截图)



(4) 使用 DDA 算法编写一个函数 dda_line，其开始应为如下所示，并检验其是否正确。

```
%DDA 画直线的算法
% P,Q 为直线段端点，pixs_line 为计算出的像素点
function pixs_line = dda_line( P , Q )

dx = Q(1)-P(1);
dy = Q(2)-P(2);
x = P(1);
y = P(2);

if abs(dx) > abs(dy)
      epls = abs(dx);
else
      epls = abs(dy);
end
```

```
xIncre = dx/epls;
yIncre = dy/epls;

n = epls + 1;
pixs_line = zeros(n,2);
pixs_line(1,1:2) = [P(1) P(2)];

for i = 2:n
    x = x+xIncre;
    y = y+yIncre;
    pixs_line(i,:) = [floor(x+0.5) floor(y+0.5)];
end
```

(5) 利用函数文件 random_line，生成 n 条随机直线段。
编写如下函数，比较两种算法的运行效率：
选取的 n：10000
Bresenham 算法运行时间：0.093860
DDA 算法运行时间：0.177811

5. 中点 Bresenham 画圆程序
(程序代码)

```
% 中点 Bresenham 画圆的算法
% R 为半径，pixs_circle 为计算出的像素点
function pixs_circle = bresenham_circle(R)
x = 0;
y = R;
d = 1-R;
while x <= y
    pixs_base(x+1,:) = [x,y]; % 第一个点
    if d<0
        d = d+2*x+3;
    else
        d = d+2*(x-y)+5;
        y = y-1;
    end
    x = x+1;
end
pixs_circle = [pixs_base;fliplr(pixs_base)];
pixs_circle = [pixs_circle;-1*pixs_circle(:,1) pixs_circle(:,2)];
pixs_circle = [pixs_circle;pixs_circle(:,1) -1*pixs_circle(:,2)];
pixs_circle = unique(pixs_circle,'rows');
pixs_circle = sortrows(pixs_circle,1);
```

(实验截图)

R=10