实验三　直线剪裁算法实验报告

学号：1525161007　姓名：秦源　班级：软件工程一班

(Cohen-Sutherland 算法)

4.1　Matlab 中 plot 命令的参数及常见图形控制命令

在上述文件末尾依次添加以下命令行，观察运行结果：

(1) legend('sin(x)','cos(x)');

(2) xlabel('angle','FontSize',12);　ylabel('value','FontSize',12);

(3) text(1.3,0.9,'sin(x)','FontName','Time New Roman');
gtext('cos(x)','FontName','Time New Roman');

(4) saveas(gcf,'myplot01.jpg');

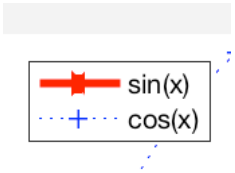
(5) figure; subplot(1,2,1); plot(x,y1.^2,'r-*'); axis([-1 10 -1 2]);

(6) subplot(1,2,2); plot(y1,y2,'b:+');

(7) axis equal;

上述命令行的作用分别是：

(1) 添加图例的标注。



(2) 在绘图窗口中的 x 轴方向上显示一个标签'angle'，y 轴方向上显示一个标签'value'。

(3)在坐标(1.3,0.9)的位置显示字符串'sin(x)'。在绘图窗口显示十字坐标系，坐标系原点随鼠标移动，在鼠标左键确定处，显示字符串'cos(x)'。

(4)存储显示的图像，命名为'myplot01.jpg'。

(5)显示新的绘图窗口。创建 1*2 的栅格，并在第一个栅格中绘制 y1^2=x；y1 = sin(x)；x=0:0.3:2*pi 的红色连续星形点的连续曲线。坐标系的大小为[-1 10 -1 2]。

(6)在第二个栅格的位置绘制单位圆，蓝色+形点。

(7) 将横轴纵轴的定标系数设成相同值 ,即单位长度相同

4.2　生成 n 条随机直线段的函数

在命令窗口中分别输入下列命令，观察运行结果：

(1) help random_lines

(2) lines = random_lines(10, [0,10], [3,5])

(3) lines = random_lines(10)

运行结果表明：

(1)

```
>> help random_lines
  build random lines with array : n*4, [x(1) y(1) x(2) y(2)]
  n is the number of lines
  Xrange(Yrange) is the range of xlabel(ylabel), default is [0,1]
```

(2)

```
>> lines = random_lines(10, [0,10], [3,5])

lines =

    8.1472    4.3115    1.5761    4.4121
    9.0579    3.0714    9.7059    3.0637
    1.2699    4.6983    9.5717    3.5538
    9.1338    4.8680    4.8538    3.0923
    6.3236    4.3575    8.0028    3.1943
    0.9754    4.5155    1.4189    4.6469
    2.7850    4.4863    4.2176    4.3897
    5.4688    3.7845    9.1574    3.6342
    9.5751    4.3110    7.9221    4.9004
    9.6489    3.3424    9.5949    3.0689
```

(3)

```
>> lines = random_lines(10)

lines =

    0.4387    0.7513    0.2760    0.8407
    0.3816    0.2551    0.6797    0.2543
    0.7655    0.5060    0.6551    0.8143
    0.7952    0.6991    0.1626    0.2435
    0.1869    0.8909    0.1190    0.9293
    0.4898    0.9593    0.4984    0.3500
    0.4456    0.5472    0.9597    0.1966
    0.6463    0.1386    0.3404    0.2511
    0.7094    0.1493    0.5853    0.6160
    0.7547    0.2575    0.2238    0.4733
```

4.4 Cohen-Sutherland 直线剪裁算法

程序代码：

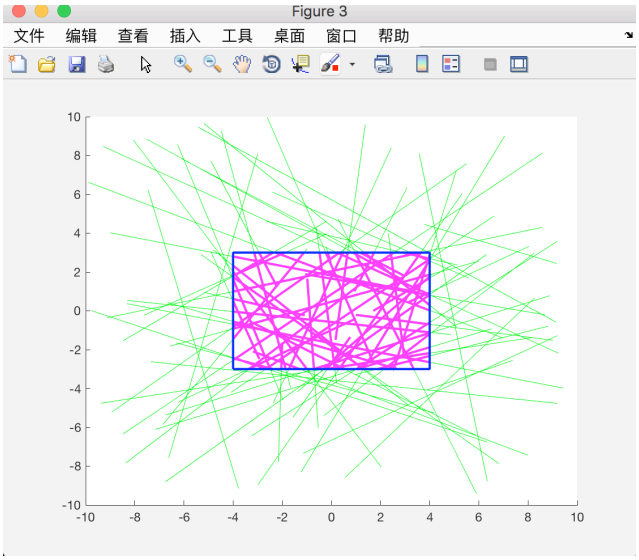
```
clear all;

n = 100;
Xmin = -4; Xmax = 4;
Ymin = -3; Ymax = 3;

figure; hold on;

P1x = rand(1,n)*20 - 10;
P2x = rand(1,n)*20 - 10;
P1y = rand(1,n)*20 - 10;
P2y = rand(1,n)*20 - 10;
P1code = zeros(n,4);
P2code = zeros(n,4);

for i = 1:n

    if P1x(i) < Xmin
```

```
        P1code(i,1) = 1;
    end
    if P1x(i) > Xmax
        P1code(i,2) = 1;
    end
    if P1y(i) < Ymin
        P1code(i,3) = 1;
    end
    if P1y(i) > Ymax
        P1code(i,4) = 1;
    end

    if P2x(i) < Xmin
        P2code(i,1) = 1;
    end
    if P2x(i) > Xmax
        P2code(i,2) = 1;
    end
    if P2y(i) < Ymin
        P2code(i,3) = 1;
    end
    if P2y(i) > Ymax
        P2code(i,4) = 1;
    end

    plot([P1x(i),P2x(i)],[P1y(i),P2y(i)],'b-');
end
hold off;

P_label = zeros(1,n);
figure; hold on;
for i = 1:n
    P_or = P1code(i,1:4) | P2code(i,1:4);
    if sum(P_or) == 0
        P_label(i) = 1;
        plot([P1x(i),P2x(i)],[P1y(i),P2y(i)],'r-');
    end
    P_and = P1code(i,1:4) & P2code(i,1:4);
    if sum(P_and) > 0
        P_label(i) = 2;
        plot([P1x(i),P2x(i)],[P1y(i),P2y(i)],'g-');
    end
end
hold off;
```

```
figure; hold on;
for i = 1:n
    if P_label(i) == 0

        P_or = P1code(i,1:4) | P2code(i,1:4);
        plot([P1x(i),P2x(i)],[P1y(i),P2y(i)],'g-');

        if P_or(1) == 1
            Py = P1y(i) + (Xmin-P1x(i))*(P2y(i)-P1y(i))/(P2x(i)-P1x(i));
            if P1x(i) < Xmin
                P1x(i) = Xmin; P1y(i) = Py;
            elseif P2x(i) < Xmin
                P2x(i) = Xmin; P2y(i) = Py;
            end
        end

        if P_or(2) == 1
            Py = P1y(i) + (Xmax-P1x(i))*(P2y(i)-P1y(i))/(P2x(i)-P1x(i));
            if P1x(i) > Xmax
                P1x(i) = Xmax; P1y(i) = Py;
            elseif P2x(i) > Xmax
                P2x(i) = Xmax; P2y(i) = Py;
            end
        end

        if P_or(3) == 1
            Px = P1x(i) + (Ymin-P1y(i))/(P2y(i)-P1y(i))*(P2x(i)-P1x(i));
            if P1y(i) < Ymin
                P1x(i) = Px; P1y(i) = Ymin;
            elseif P2y(i) < Ymin
                P2x(i) = Px; P2y(i) = Ymin;
            end
        end

        if P_or(4) == 1
            Px = P1x(i) + (Ymax-P1y(i))/(P2y(i)-P1y(i))*(P2x(i)-P1x(i));
            if P1y(i) > Ymax
                P1x(i) = Px; P1y(i) = Ymax;
            elseif P2y(i) > Ymax
                P2x(i) = Px; P2y(i) = Ymax;
            end
        end
        if P1x(i) >= Xmin & P1x(i) <= Xmax & ...
```

```
                P2x(i) >= Xmin & P2x(i) <= Xmax & ...
                P1y(i) >= Ymin & P1y(i) <= Ymax & ...
                P2y(i) >= Ymin & P2y(i) <= Ymax
            plot([P1x(i),P2x(i)],[P1y(i),P2y(i)],'m-','LineWidth',2);
        end
        plot([-4 4 4 -4 -4],[-3 -3 3 3 -3],'b-','LineWidth',2);
    end
end
hold off;
```

运行结果截图：



(Cyrus-Beck 算法)

4.4 计算多边形各边的内法向量

在命令窗口中依次输入以下语句，运行上述函数，并观察运行结果：

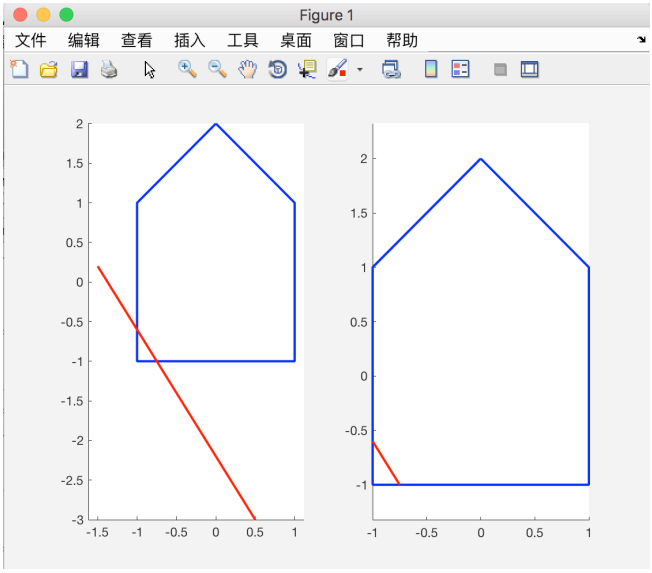
(1) Nv = innerNormVect([0 0; 1 0; 1 1; 0 1])

(2) help norm

(3) Nv = innerNormVect([0 0 0; 1 1 1; 1 0 0])

运行结果：

(1) 运行结果为：




```
>> Nv = innerNormVect([0 0; 1 0; 1 1; 0 1])

Nv =

     0     1
    -1     0
     0    -1
     1     0
```

(2) norm 函数的作用为：

```
>> help norm
norm - 矢量范数和矩阵范数

    此 MATLAB 函数 返回矢量 v 的 2 范数或欧几里得范数。

    n = norm(v)
    n = norm(v,p)
    n = norm(X)
    n = norm(X,p)
    n = norm(X,'fro')
```

(3) 运行结果为：

```
>> Nv = innerNormVect([0 0 0; 1 1 1; 1 0 0])
错误使用 innerNormVect (line 5)
The polygonal vertex should be [x1 y1; x2 y2; ...] !
```

(Liang-Barsky 算法)

4.6 Liang-Barsky 算法

程序代码：

```
clear all

pixs_line=LBLineClip(3,15,7,19,0,0,18,18);
draw_graphics(pixs_line);

function [max,min,i]=LBLineClipTest(p,q,umax,umin)
r=0;
```

```
if p<0 % p 小于 0 时比较最大值
    r=q/p;
    if r>umin
        i=0; % umax 小于 umin 直线段才有在窗口的部分
        max=umax;
        min=umin;
        return;
    elseif r>umax
            umax=r;
    end

elseif p>0 p 大于 0 时比较最大值
    r=q/p;
    if r<umax
        i=0; % umax 小于 umin 直线段才有在窗口的部分
        max=umax;
        min=umin;
        return;
    elseif r<umin
        umin=r;
    end

elseif q<0
    i=0; %处理 p=0 的情况
    max=umax;
    min=umin;
    return;
end
    max=umax;
    min=umin;
    i=1;
    return;
end

function pixs_line=LBLineClip(xwl,xwr,ywb,ywt,x1,y1,x2,y2)
deltax=x2-x1;
deltay=y2-y1;
umax=0;
umin=1;
[umax,umin,i1]=LBLineClipTest(-deltax,x1-xwl,umax,umin);
if i1 == 1
    [umax,umin,i2]=LBLineClipTest(deltax,xwr-x1,umax,umin);
    if i2==1
        [umax,umin,i3]=LBLineClipTest(-deltay,y1-ywb,umax,umin);
```

```
        if i3==1
            [umax,umin,i4]=LBLineClipTest(deltay,ywt-y1,umax,umin);
            if i4==1
                new_x1=floor(x1+umax*deltax+0.5);
                new_y1=floor(y1+umax*deltay+0.5);
                new_x2=floor(x1+umin*deltax+0.5);
                new_y2=floor(y1+umin*deltay+0.5);

                P=[new_x1 new_y1];
                Q=[new_x2 new_y2];
                pixs_line = bresenham_line(P,Q);
                return;
            end
        end
    end
end
pixs_line=[];
end
```

运行结果截图：