《机床数控技术》课程设计

说明书

班级：0519108

学号：051910806

姓名：樊俊伟

指导教师：缪群华

时间：2022 年 6 月

目录

一、课程设计介绍

1.1 课程设计任务

（1）直线插补：DL2　DDA 法插补第二象限直线
（2）圆弧插补：PA32　逐点比较法插补第三象限到第二象限顺圆弧

课程设计实际完成情况：DDA 法与逐点法插补全象限直线、DDA 法和逐点法插补全象限圆弧。

1.2 课程设计任务要求

（1）具有必要的数据输入界面，如起点、终点、圆心、半径及插补步长等；
（2）具有插补过程的动态显示功能，如单步插补、连续插补；
（3）插补的步长可调。

1.3 编程软件

MATLAB 2018b

1.4 功能说明

（1）具有数据输入界面，可输入直线插补的起点、终点，圆弧插补的起点、终点、圆心坐标，插补的步长。

（2）具有插补过程的动态显示功能，如单步插补、连续插补。

（3）直线的起点、圆弧的圆心在坐标系中的位置可变（即直线的起点、圆弧的圆心可不设定在坐标原点）。

（4）具有清除数据，清楚图像，退出程序功能。

（5）可进行全象限逐点比较法插补，包括全象限逐点法直线插补和全象限顺逆圆弧逐点法插补。

（6）可进行全象限 DDA 法直线插补和圆弧插补，寄存器位数可调可以选择左移规格化和余数寄存器预 置数的方法提高 DDA 法直线插补速度与质量。

（7）插补的步长可调，连续插补速度可调。

（8）保存插补轨迹为图片的功能。

界面截图：

图 1 界面截图

二、程序原理

2.1 逐点法原理

逐点比较法又称代数运算法或醉步法，是早期数控机床开环系统中广泛采用的一种插补方法，可实现直线插补、圆弧插补，也可用于其他非圆二次曲线（如椭圆、抛物线和双曲线等）的插补。

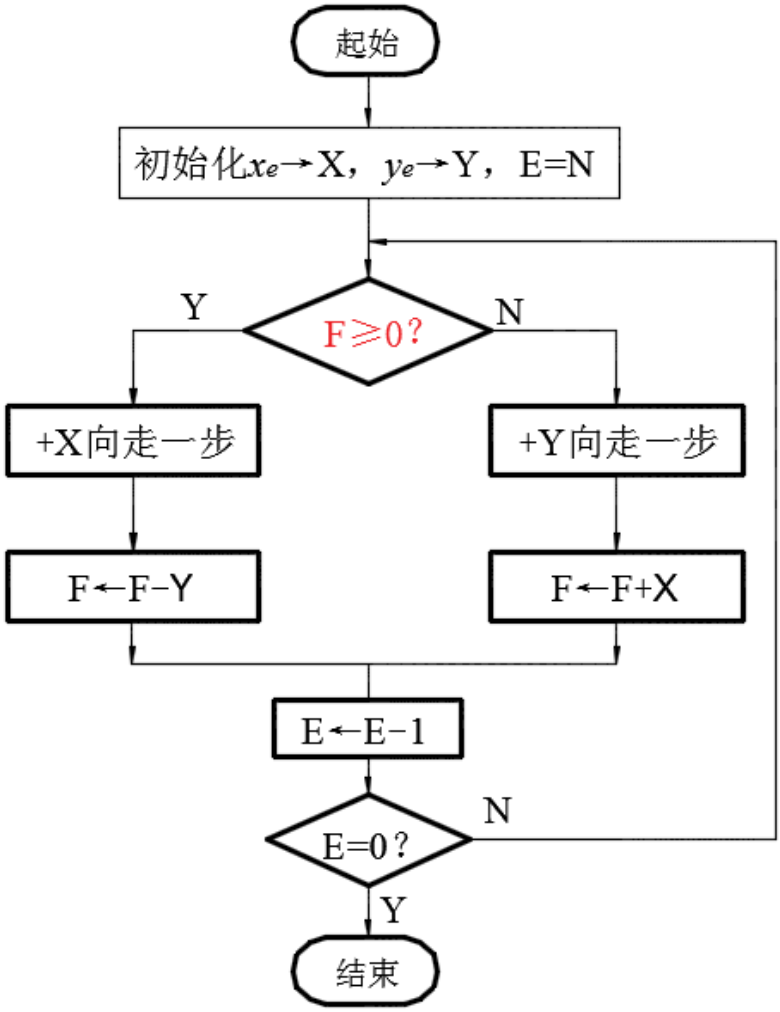
逐点法的基本原理是本次仅向一个坐标轴输出一个进给脉冲，每走一步都要将加工动点坐标与理论加工轨迹相比较，判断实际加工点与理论加工轨迹的偏移位置，通过偏差函数计算二者之间的偏差，从而决定下一步的进给方向。每进给一步都要完成偏差判别，坐标进给，偏差计算和终点判别四个工作节拍。

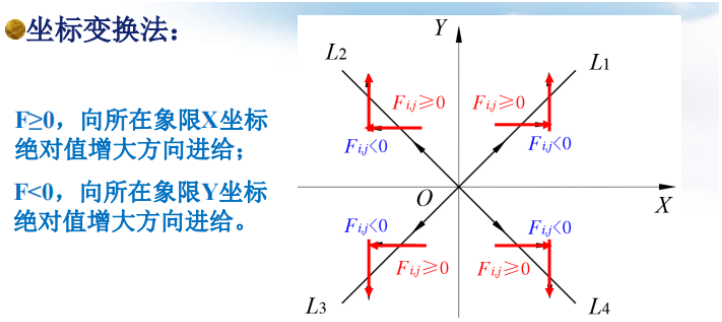
2.1.1 逐点法直线插补原理

对于动点 P(xi,yi),存在三种情况：加工点 P 在直线上，F=0；加工点 P 在直线上方，F>0;加工点 P 在直线下方，F<0；F≥0，向所在象限 X 坐标 绝对值增大方向进给；F<0，向所在象限 Y 坐标绝对值增大方向进给。

不同象限的插补采用坐标变换法完成即坐标用绝对值进行计算，不同象限 x 与 y 的进给方向有所区别。

例如第一象限插补流程图如下：





	I	II	III	IV
$F_{i,j} \geq 0$	+X	-X	-X	+X
$F_{i,j} < 0$	+Y	+Y	-Y	-Y

2.1.2 逐点法圆弧插补原理

逐点法圆弧插补与直线插补类似，只不过偏差判别函数有所区别。统一于第一象限的逆圆弧插补公式，点的坐标采用的是绝对值。

插补第一象限逆圆弧 AB，其圆心位于原点 O（0，0），半径为 R，令加工点的坐标为 P（xi，yj），则逐点比较法圆弧插补的偏差判别函数为：

$$F_{i,j} = x_i^2 + y_j^2 - R^2$$

例如插补第一象限逆圆弧

$F_{i,j} \geq 0$ 时

$$\begin{aligned} F_{i+1,j} &= x_{i+1}^2 + y_j^2 - R^2 \\ &= (x_i - 1)^2 + y_j^2 - R^2 \\ &= (x_i^2 + y_j^2 - R^2) - 2x_i + 1 \\ &= F_{i,j} - 2x_i + 1 \end{aligned}$$

$F_{i,j} \leq 0$ 时

$$\begin{aligned} F_{i,j+1} &= x_i^2 + y_{j+1}^2 - R^2 \\ &= x_i^2 + (y_j + 1)^2 - R^2 \\ &= (x_i^2 + y_j^2 - R^2) + 2y_j + 1 \\ &= F_{i,j} + 2y_j + 1 \end{aligned}$$

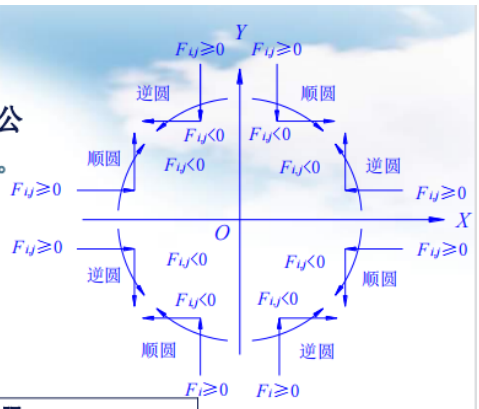
不同象限插补采用坐标变换法进行插补

2）坐标变换法

统一于第一象限的逆圆弧插补公式，点的坐标采用的是绝对值。

$F_{i,j} \geq 0$ $|x_{i+1}| = |x_i| - 1$ $F_{i+1,j} = F_{i,j} - 2|x_i| + 1$

$F_{i,j} < 0$ $|y_{j+1}| = |y_j| + 1$ $F_{i,j+1} = F_{i,j} + 2|y_j| + 1$



根据象限确定进给方向

图形	脉冲	象限			
		Ⅰ	Ⅱ	Ⅲ	Ⅳ
G03	Δx	-X	-Y	+X	+Y
	Δy	+Y	-X	-Y	+X
G02	Δx	-y	+X	+Y	-X
	Δy	+X	+Y	-X	-Y

2.2 DDA 法原理

与逐点比较法类似，数字积分法就是一种脉冲分配的方法，利用待插补的直线或圆弧或其它函数曲线的特征（起点、终点、圆心、半径等），采用一种算法分配脉冲，实现坐标进给，最终走出符合精度要求的轨迹。DDA 插补原理如下：

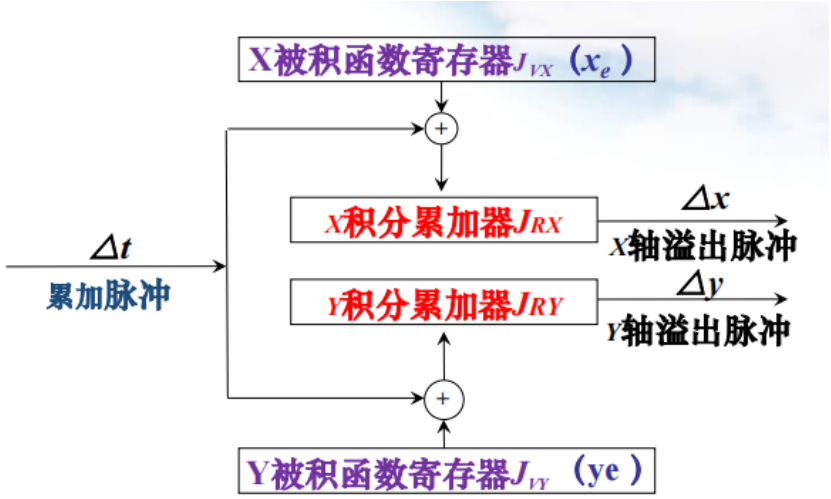
2.2.1DDA 直线插补原理

以第一象限为例直线的起点在原点 0（0,0），终点为 A（xe ， ye）。用软件实现数字积分法直线插补时在内存中设立几个存储单元 分别存放 x e 及其累加值 Σ x e 和 y e 及其累加值 Σ y e 在每次插补运算循环过程中进行以下求和运算，用运算结果溢出的脉冲△x 和△ y 来控制机床进给，就可走出所需的直线轨迹。

数字积分法插补其他象限的直线时一般 将其他各象限直线的终点坐标均取绝对值这样 它们的插补计算公式和插补流程图与插补第一象限直线时一样 而脉冲进给方向总是直线终点坐标绝对值增加的方向 。

$$\Sigma x_e + x_e \rightarrow \Sigma x_e$$

$$\Sigma y_e + y_e \rightarrow \Sigma y_e$$

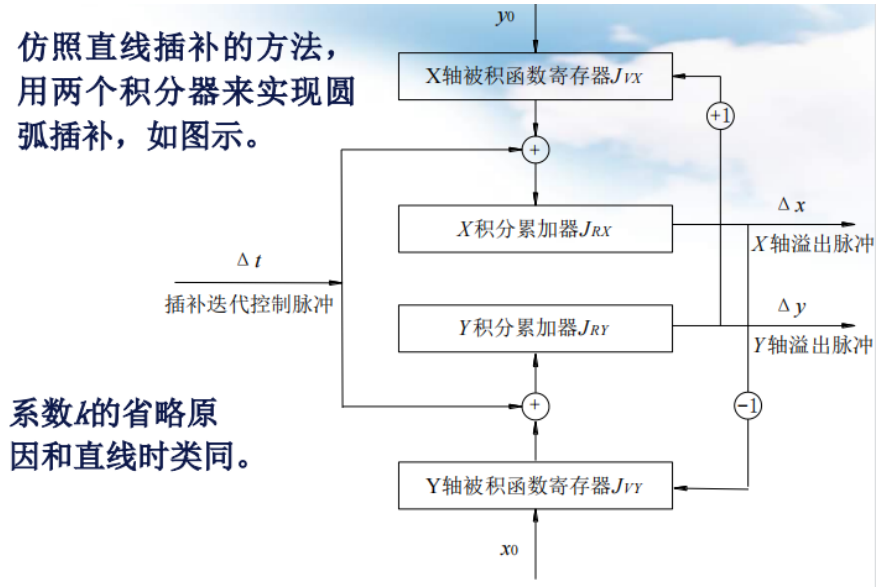


2.2.2 DDA 圆弧插补原理

类似于 DDA 直线插补原理，但是有所区别。直线插补时，被积函数寄存器 Jvx 、 Jvy 分别存放常值（终点坐标） xe 、 ye ，而圆弧插补时动点坐标的存放恰好位置对调，即 y 存入 Jvx x 存入 J vy ，且在起点时 Jvx Jvy 分别寄存起点坐标值 y0 x0 。圆弧插补终止：当溢出到 x 轴的脉冲数为 xe x0 ||，溢出到 y 轴的脉冲数为 ye y0 时就停止积分运算，而直线 DDA 是累加 m=2 n 次结束。

对于其它象限的顺圆、逆圆插补运算过程和积分器结构基本上与第一象限逆圆弧是一致的，但区别在于，控制各坐标轴的△x、△y 的进给方向不同，以

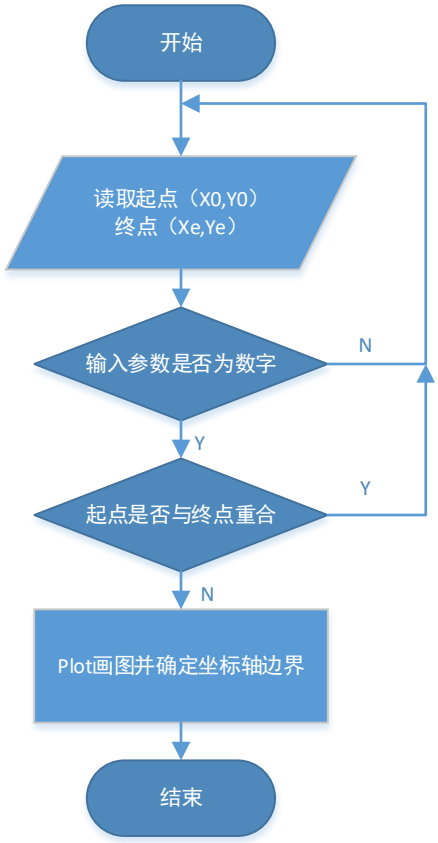
及修改 Jvx、Jvy 内容时是加“1”还是减“1”，要由 xi 和 yj 坐标值的增减而定。



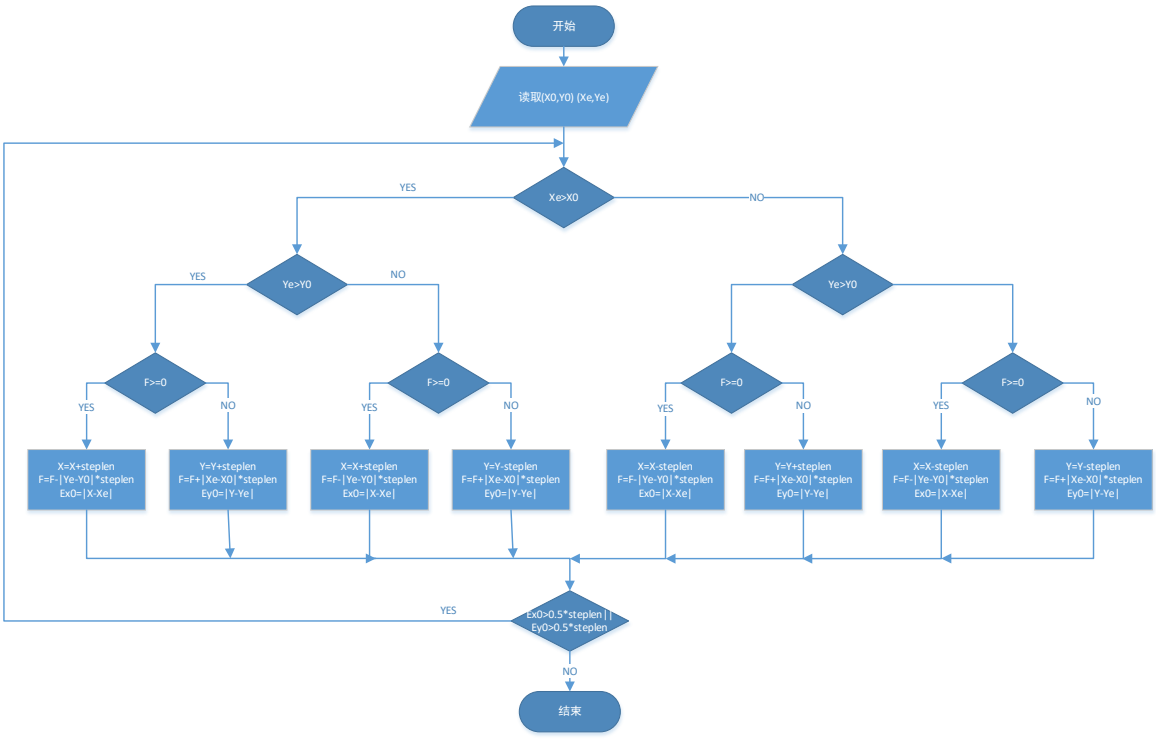
三、程序设计流程图

3.1 直线插补

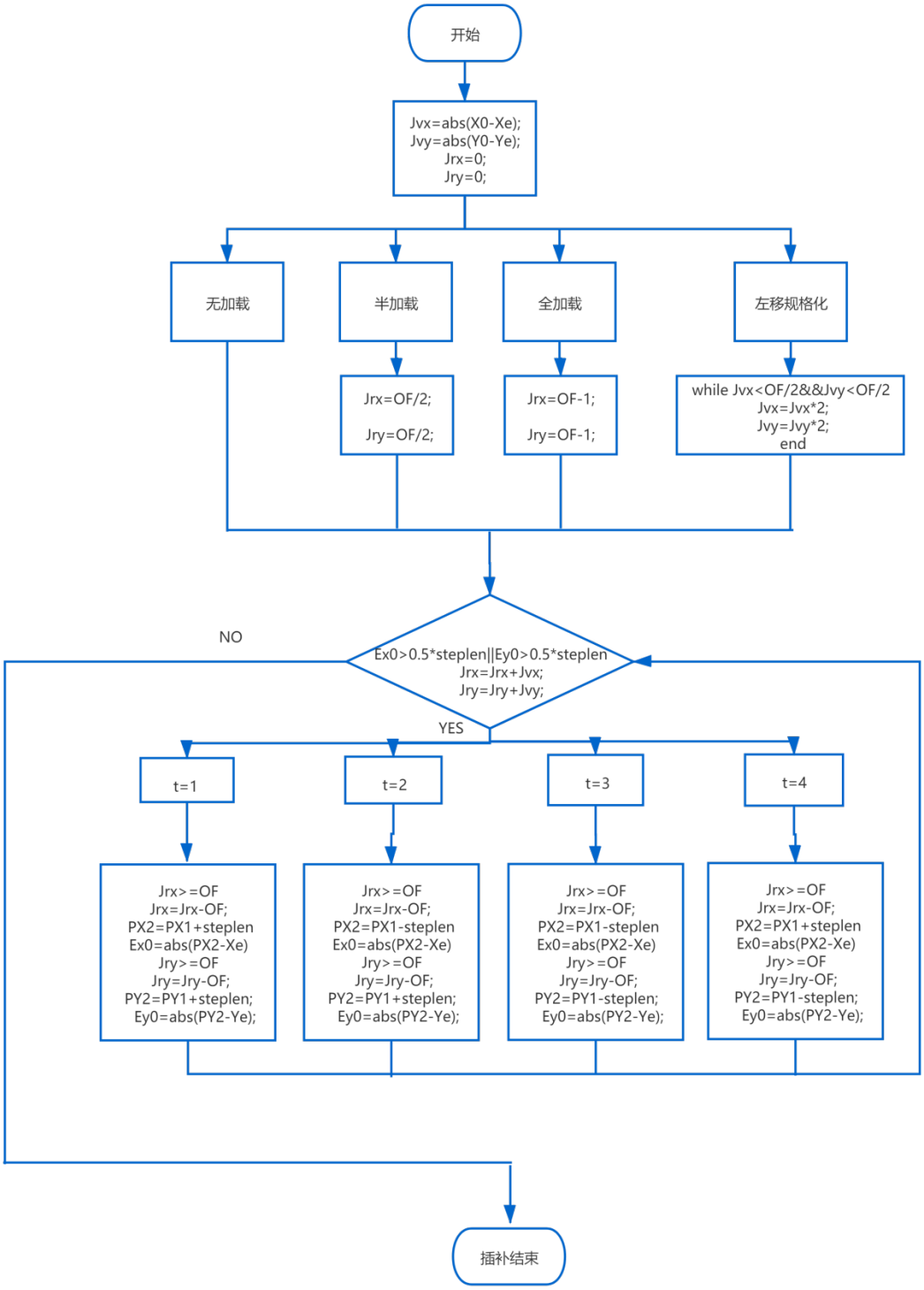
3.1.1 理论直线绘制程序流程图



3.1.2 逐点法直线插补绘制程序流程

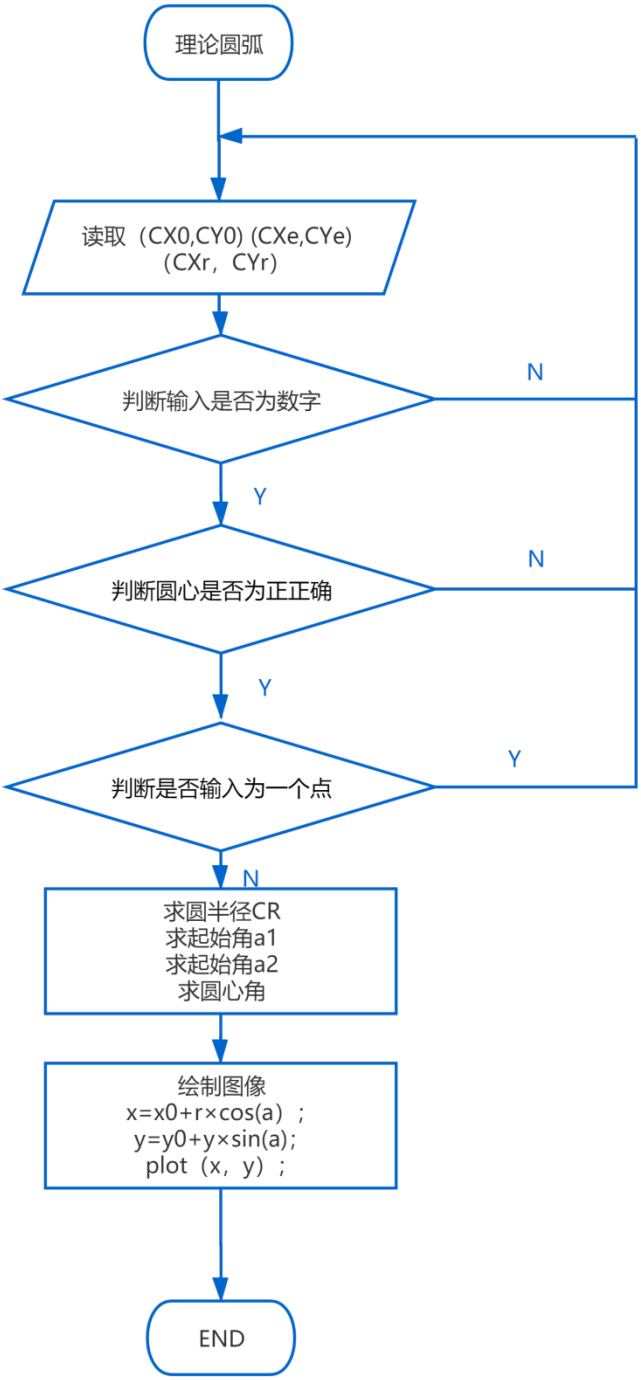


3.1.3 DDA 法直线插补绘制程序流程图

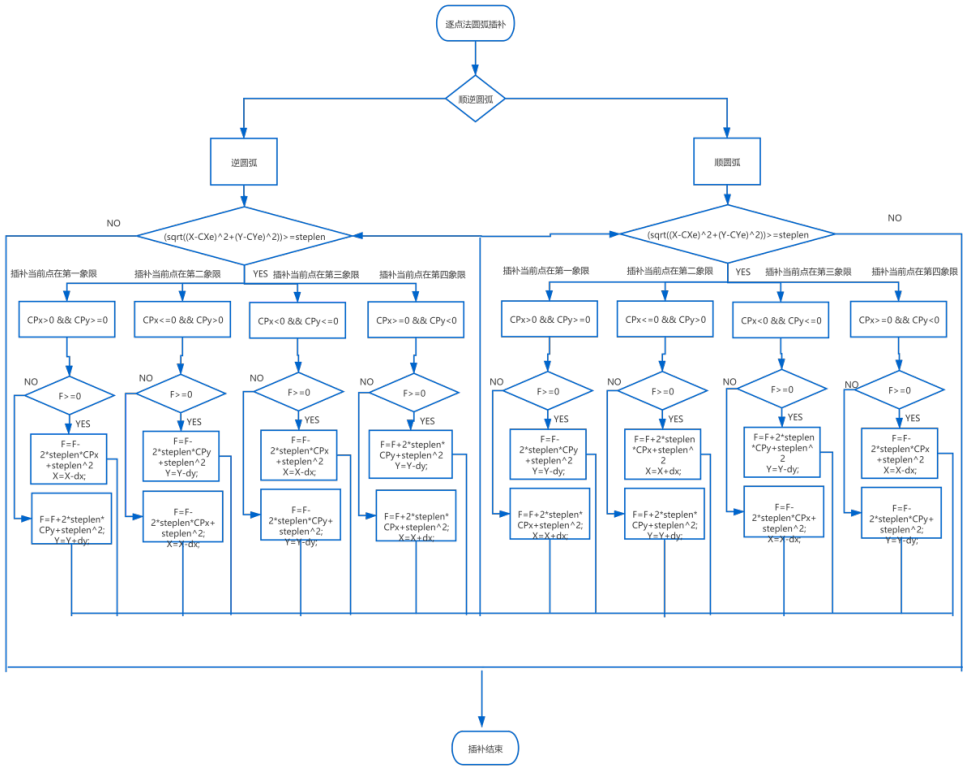


3.2 圆弧插补

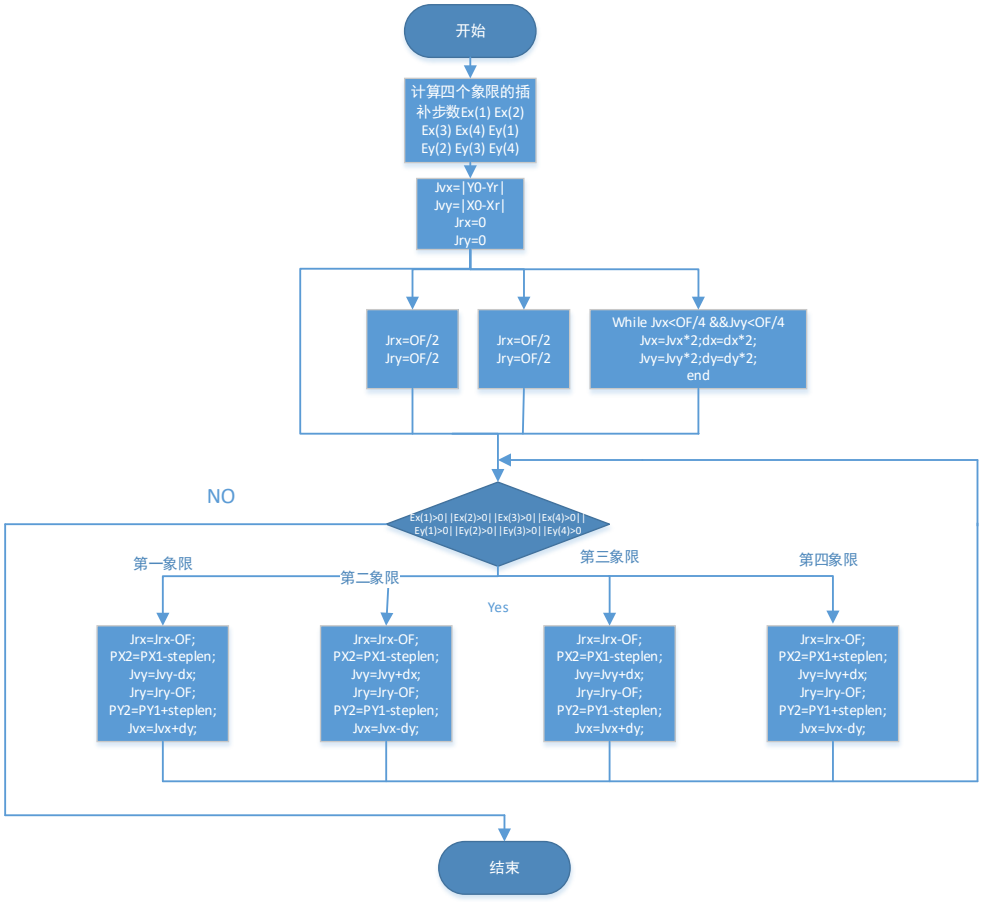
3.2.1 理论圆弧绘制程序流程图



3.2.2 逐点法圆弧插补绘制程序流程图



3.2.3 DDA 法圆弧插补绘制程序流程图



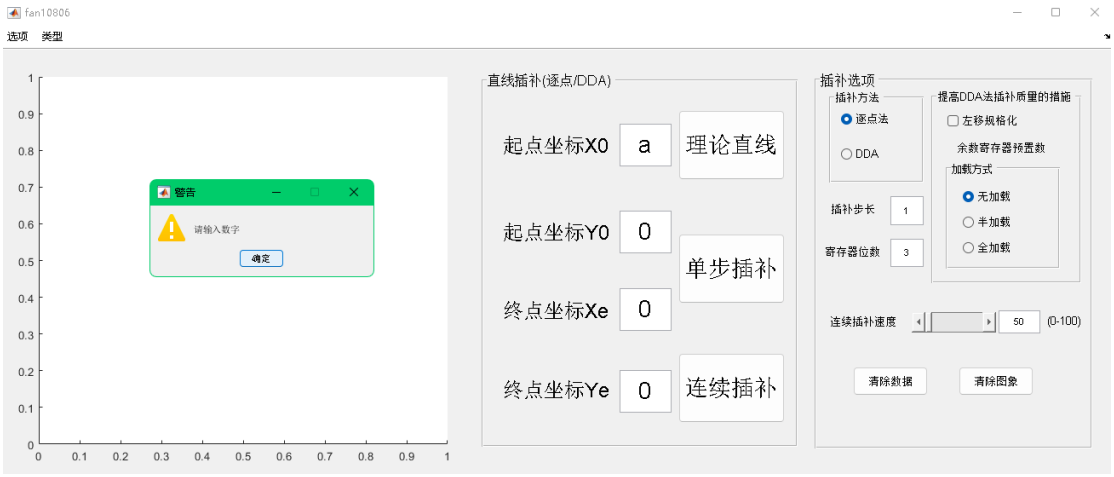
四、说明及展示

整体说明，本程序可分别通过窗口与用户交互完成插补数据的输入；后根据输入的数据选择自动完成圆弧、直线的全象限逐点法插补、直线的全象限 DDA 法插补（包含左移规格化、加载功能），插补过程中可选择速度可调的连续插补或单步插补。

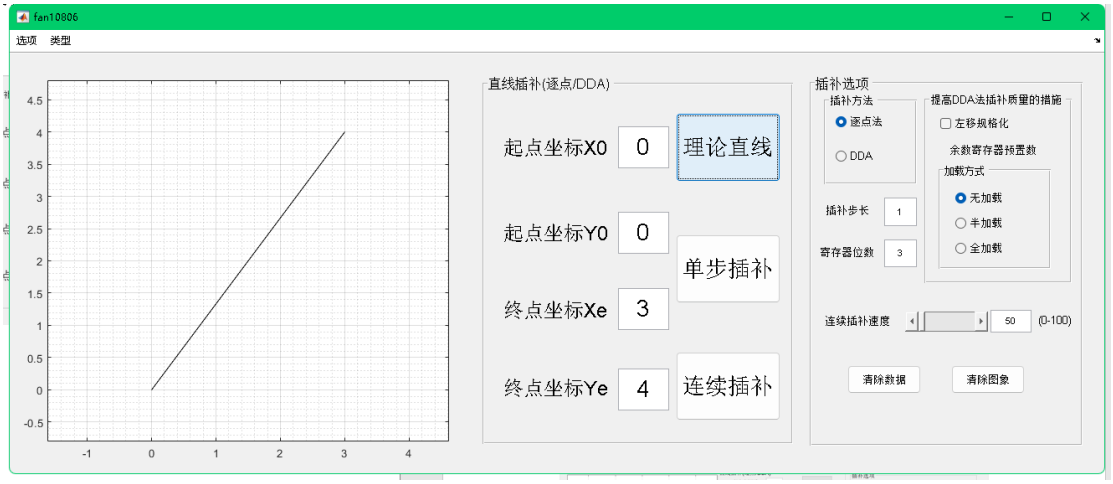
4.1 直线插补

4.1.1 理论直线绘制

由数据直接绘制理论直线，数据可为小数，整数，如果不是数字或者没有输入数据会提示输入数字。



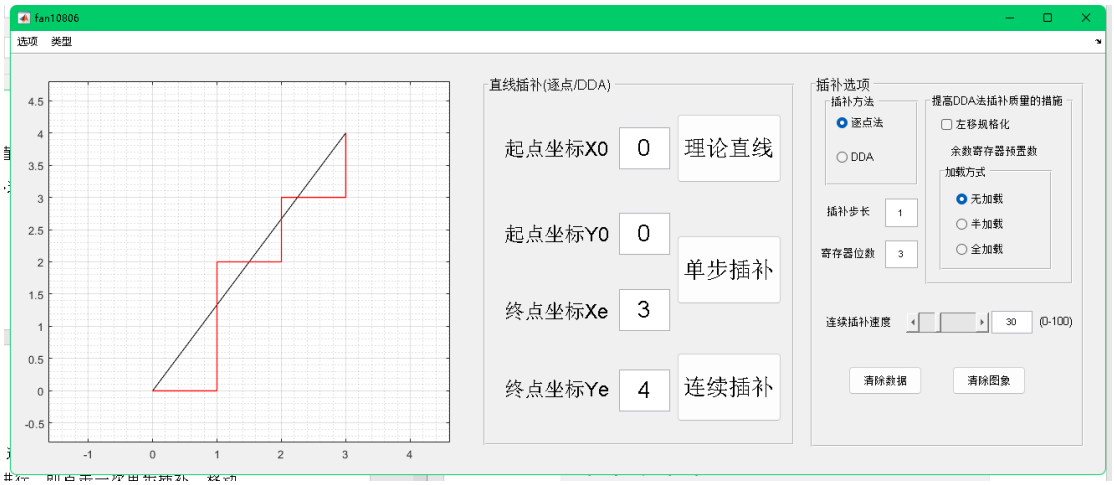
输入不是数字报错



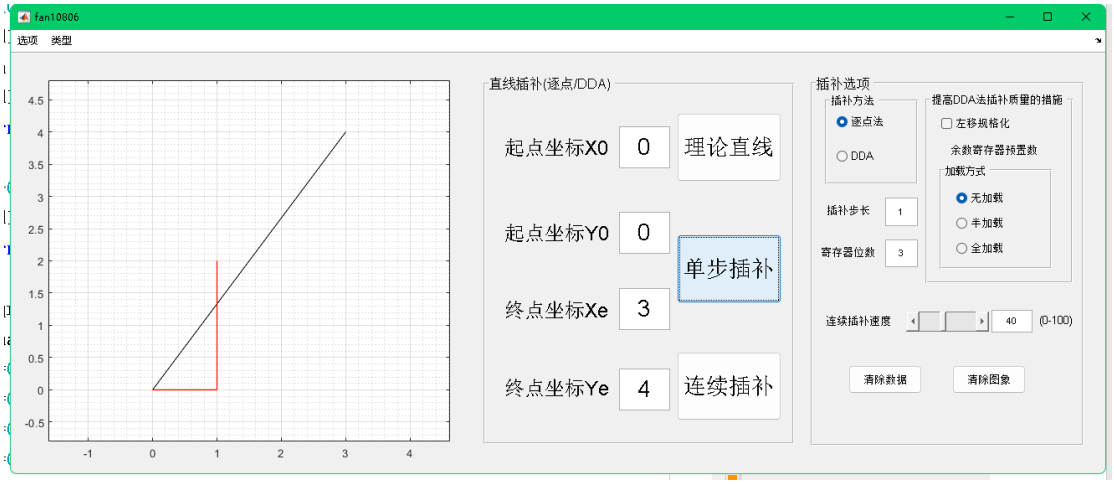
理论直线绘制

4.1.2 逐点比较法直线插补

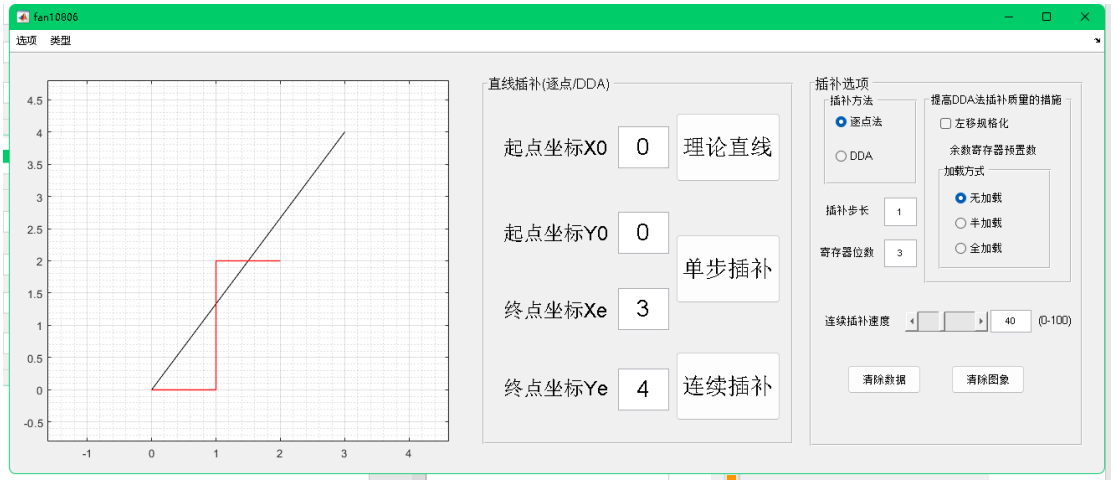
输入正确的起点终点数据，先进行理论直线的绘制，设置插补步长（不能为小于等于 0 的数，不然会报错），设置插补速度（滑条和输入框均可设置速度），点击连续插补，插补路径将会以设定的速度显示出来，如图中的红色路径。接着点击单步插补，插补将会一步一步进行，即点击一次单步插补，移动一个步长的距离。



逐点比较法连续插补



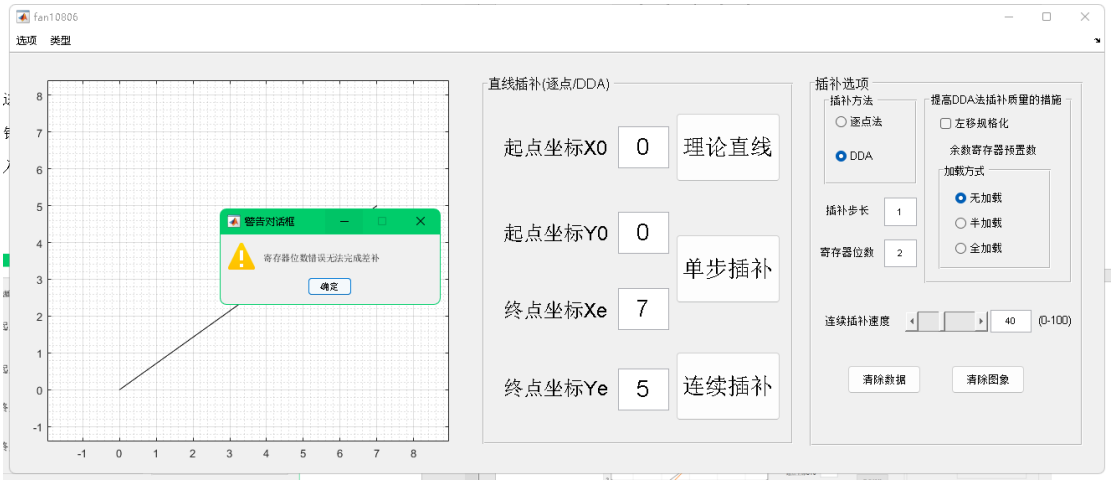
逐点比较法单步插补部分一



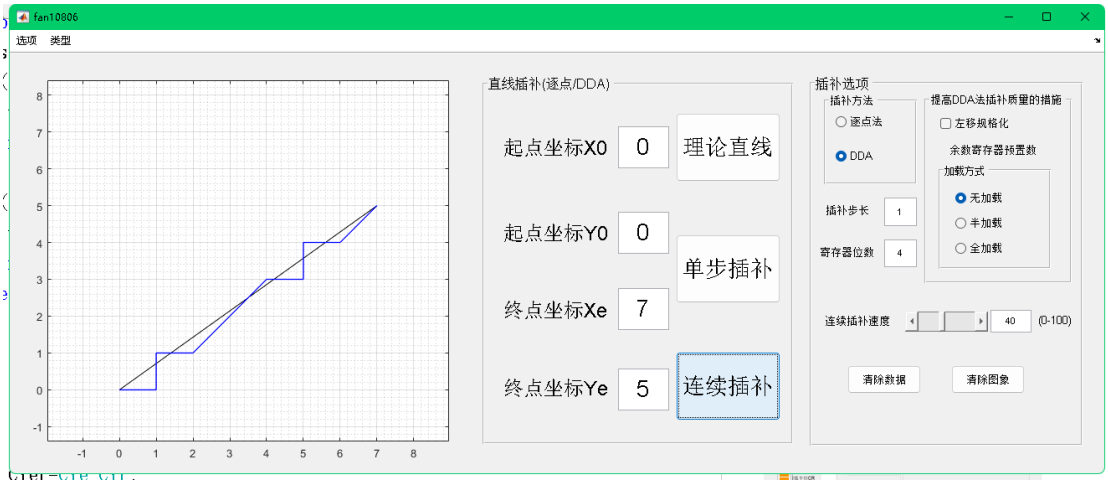
逐点比较法单步插补部分二

4.1.3 DDA 法直线插补

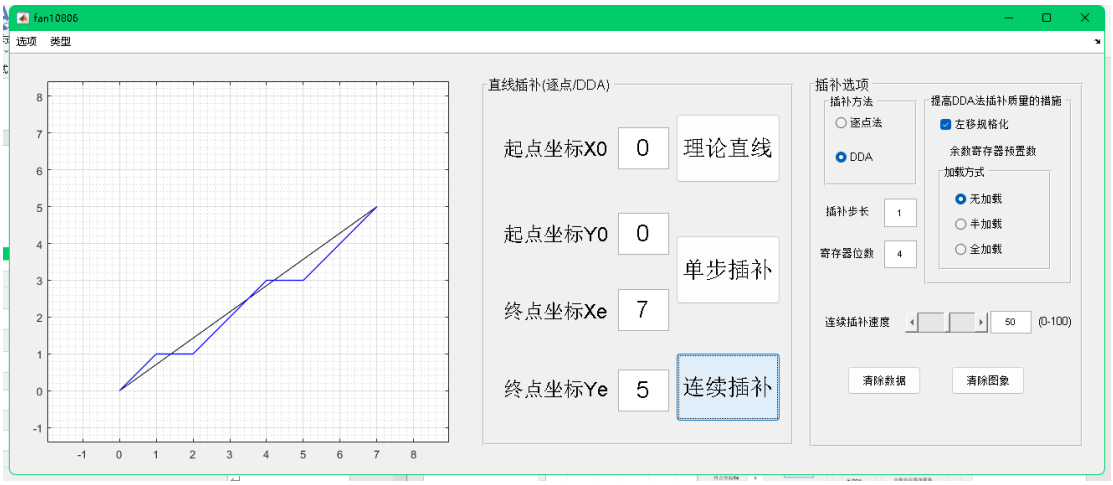
根据理论直线，可对全象限分别进行加载，半加载，全加载插补，以及左移规格化。若步长小于等于 0 将提示错误。系统自动计算需要的寄存器位数并显示于屏幕上，也可以由用户自行输入，但输入后插补执行前系统仍然会自动检验输入的寄存器位数是否足够。



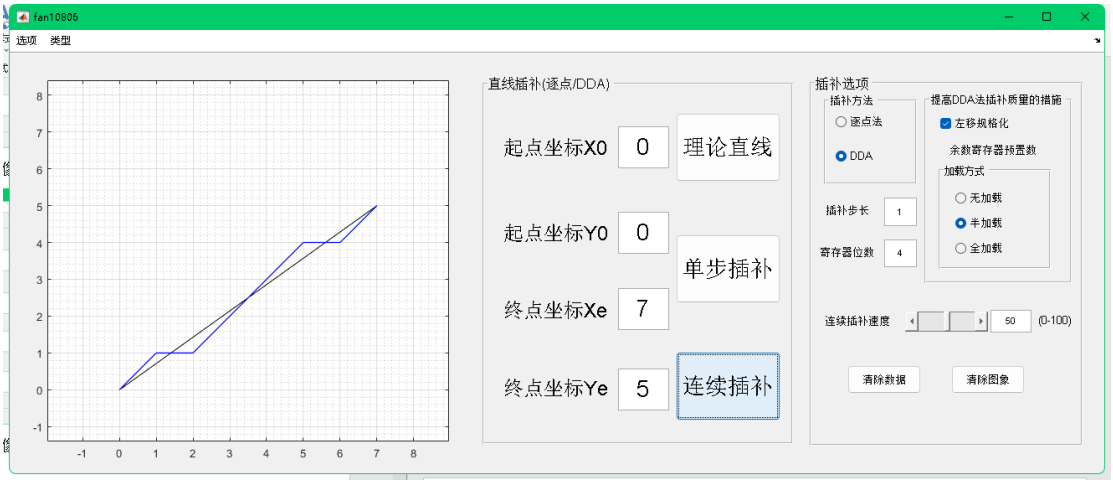
寄存器位数不够报错



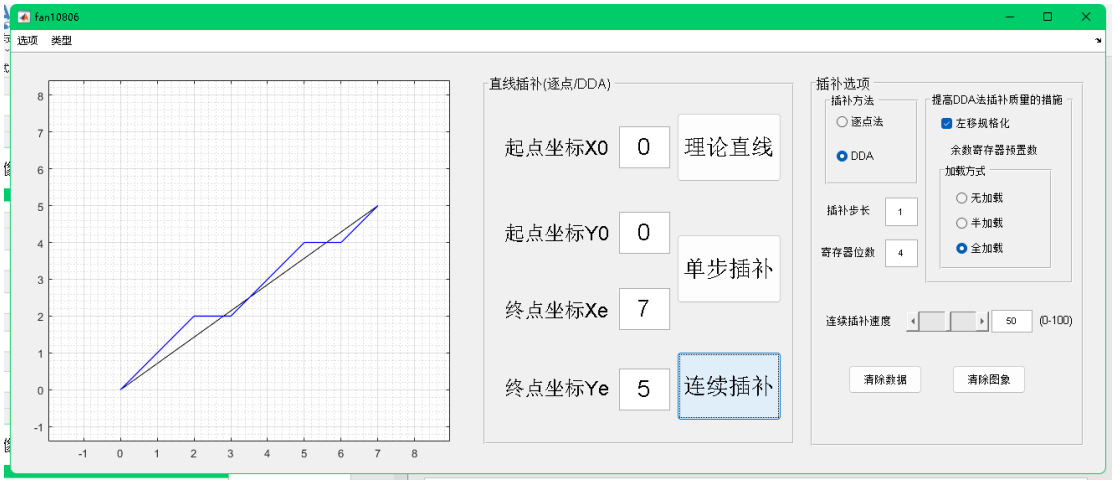
未左移规格化图像



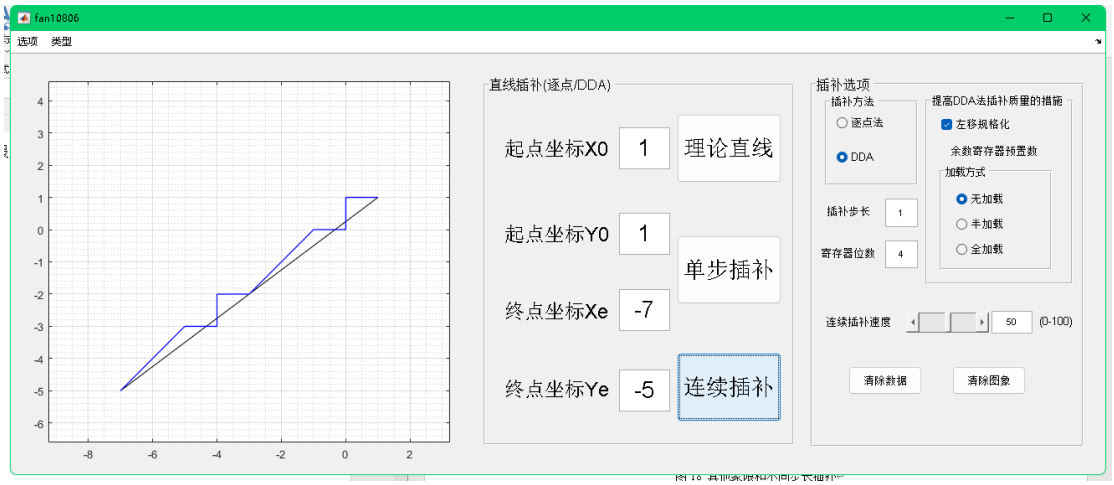
左移规格化后图像



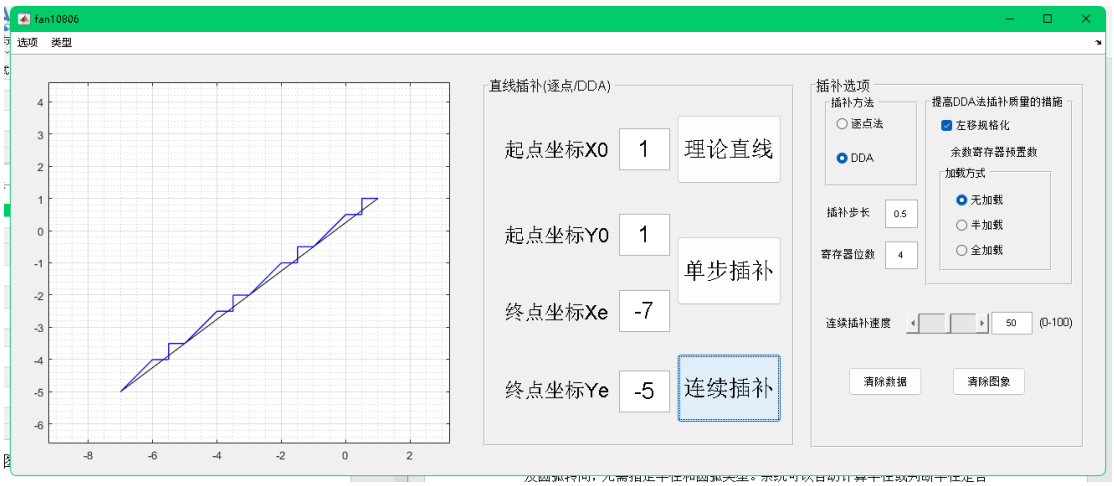
半加载后图像



全加载后图像



其他现象直线插补图像



步长可调图像

4.2 圆弧插补

对于圆弧的插补，用户从圆弧插补界面输入圆弧起点、终点和圆心坐标，

以及圆弧转向，无需指定半径和圆弧类型。系统可以自动计算半径或判断半径是否有误。信息验证符合后可绘制圆弧并将圆弧半径显示出来，再根据用户进一步指令对已绘制圆弧进行单步插补或者连续插补。

4.2.1 理论圆弧绘制

根据用户输入的圆弧信息，将会提示输入参数是否为数字，也会对所输入的信息进行判断。若输入数据格式无误则可进行理论圆弧绘制。

输入数据错误报错图

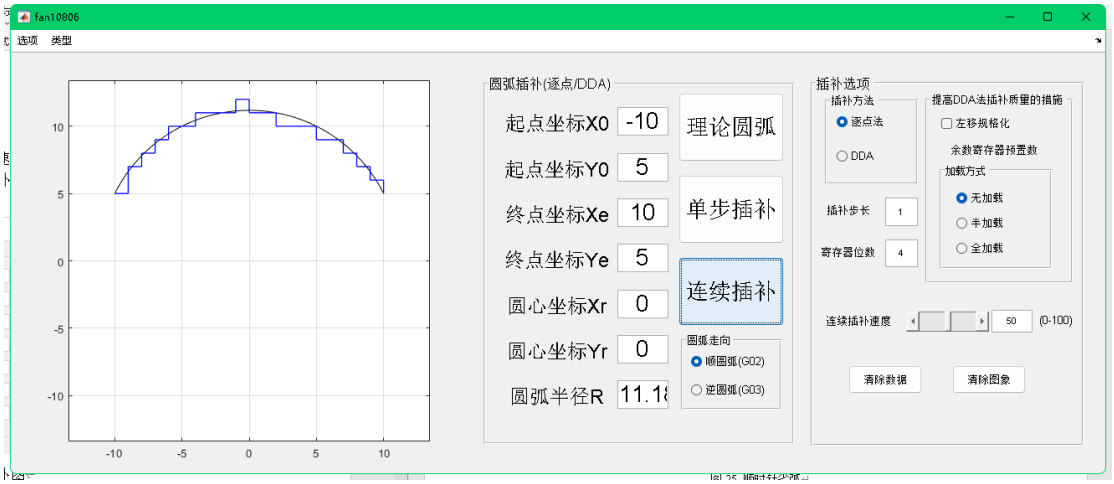
优弧绘制图

劣弧绘制图

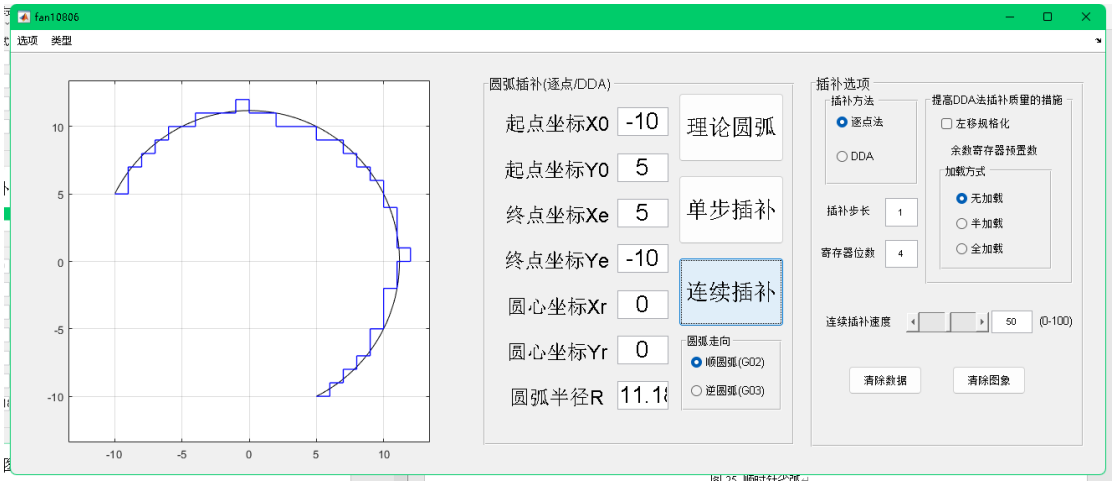
4.2.2 逐点比较法圆弧绘制

逐点比较法圆弧插补时可任意更改步长与速度进行插补，程序会自动判断该步长是否能够无误差完成圆弧起点到终点的插补，若无法完全到达终点会报错且拒绝插补。

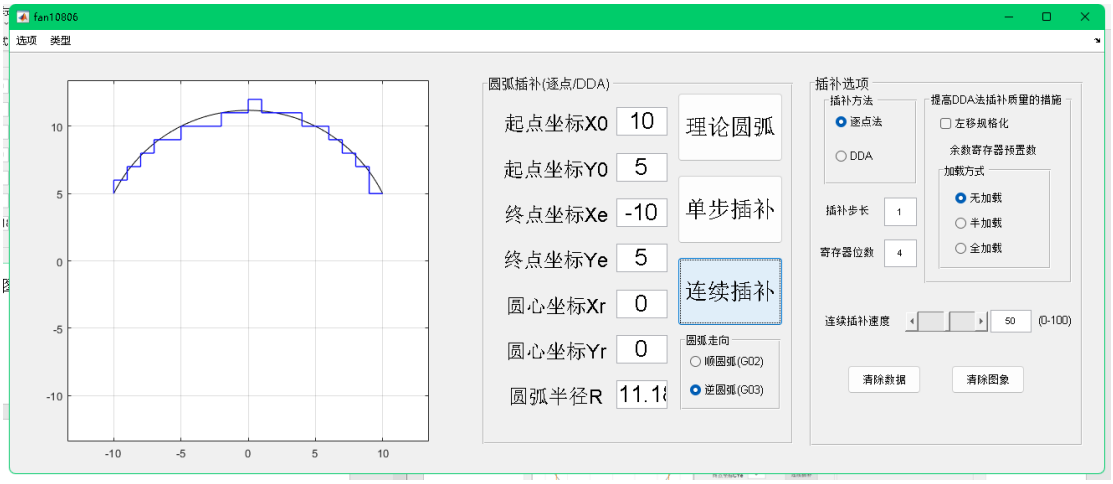
部分步长无法插补图



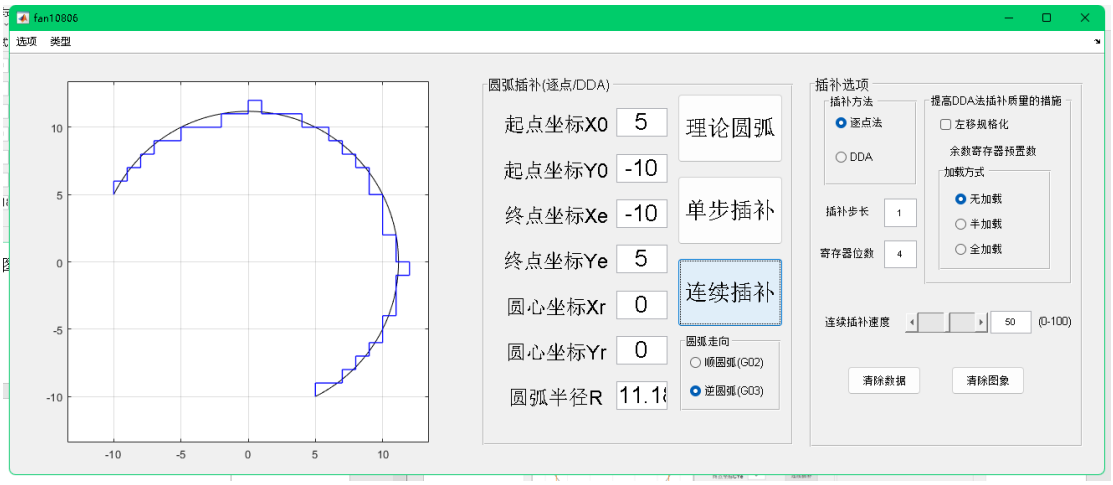
插补顺时针劣弧图



插补优顺时针弧图



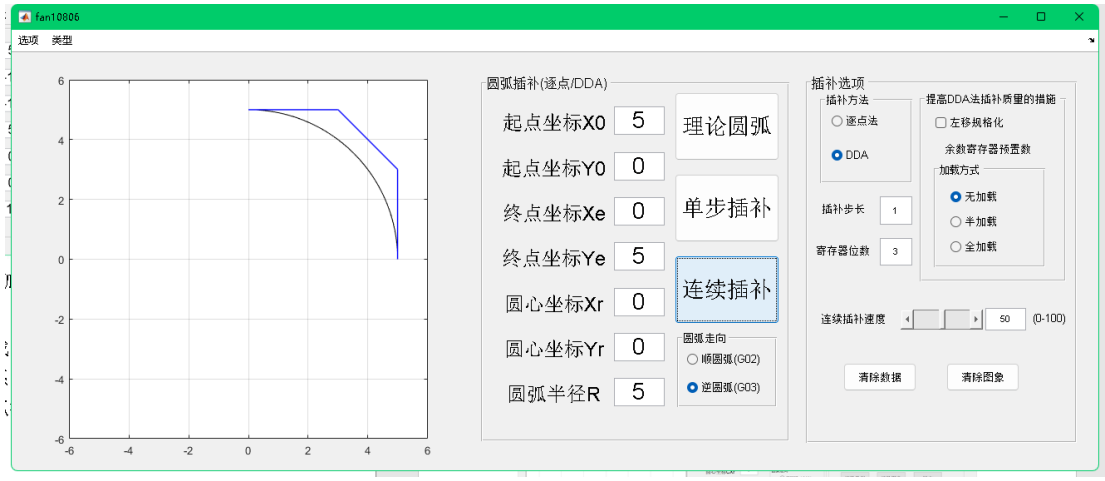
插补逆时针劣弧图



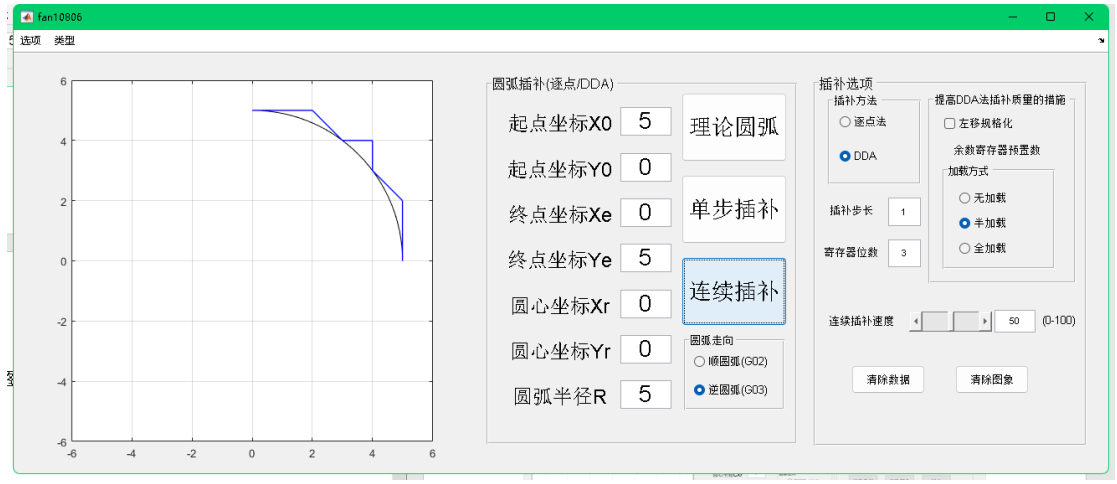
插补逆时针优弧图

4.2.3 DDA 法圆弧插补

根据理论圆弧，可对全象限分别进行加载，半加载，全加载插补，以及左移规格化。若步长小于等于零将提示错误。系统自动计算用户输入的的寄存器位数是否符合要求，若符合要求才会进行圆弧插补。

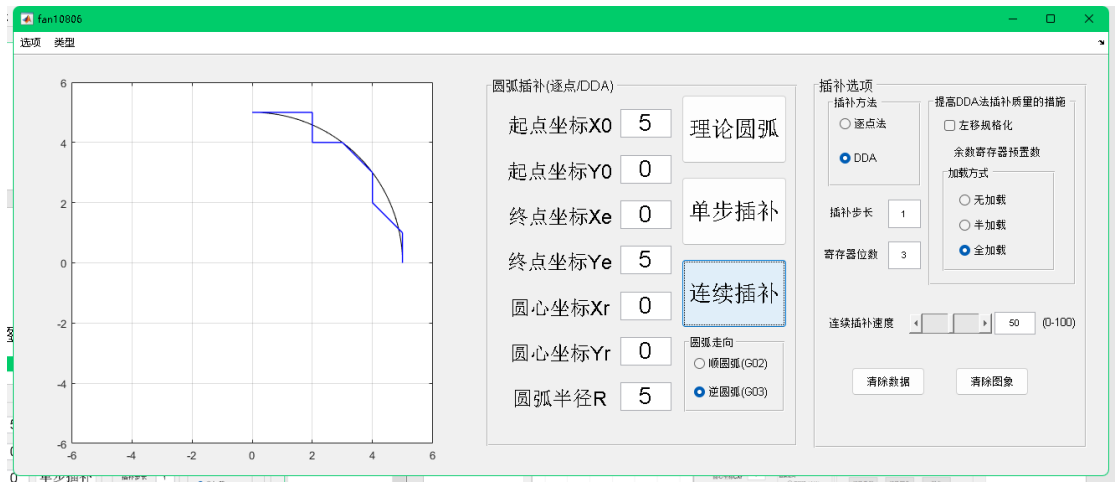


无加载插补图

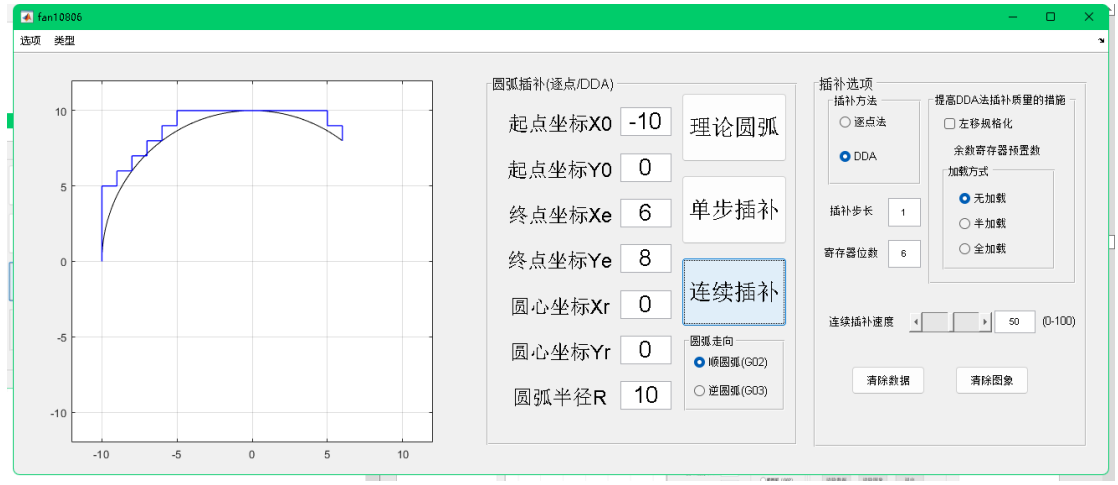


半加载插补图

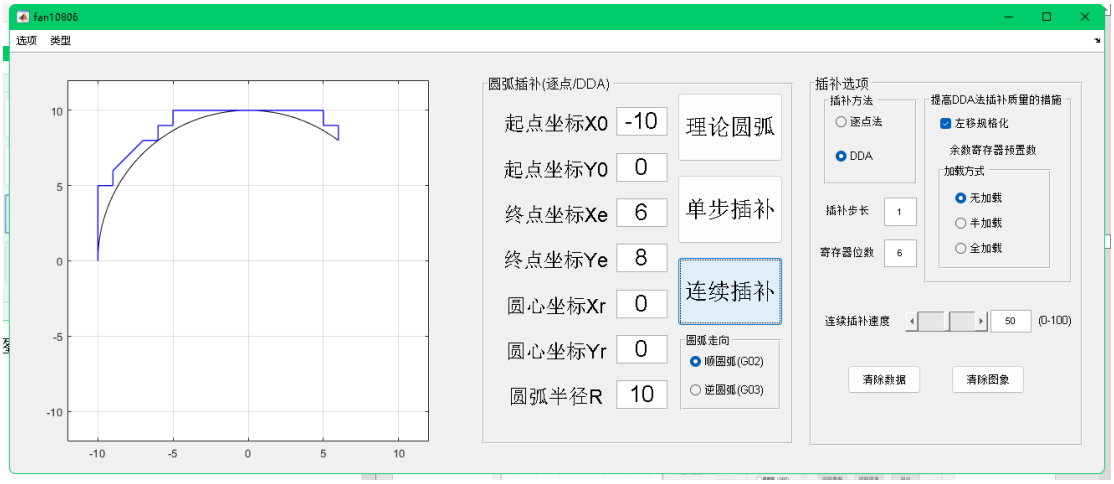
.



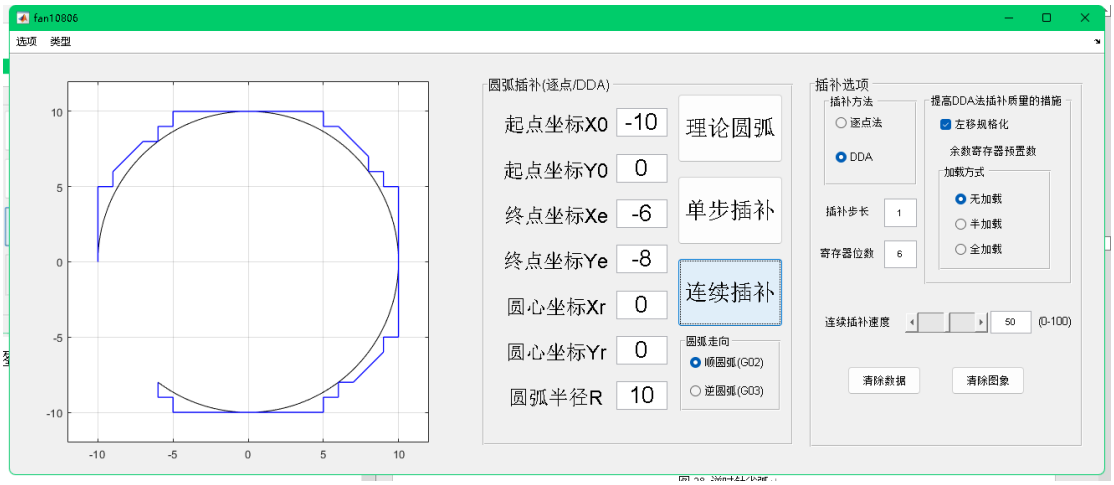
全加载插补图



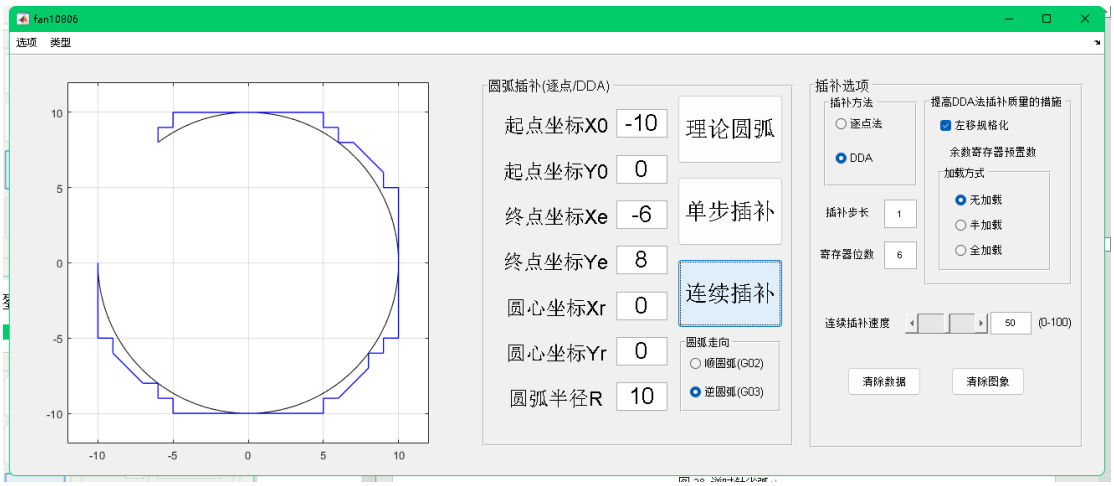
未左移规格化插补图



左移规格化后插补图

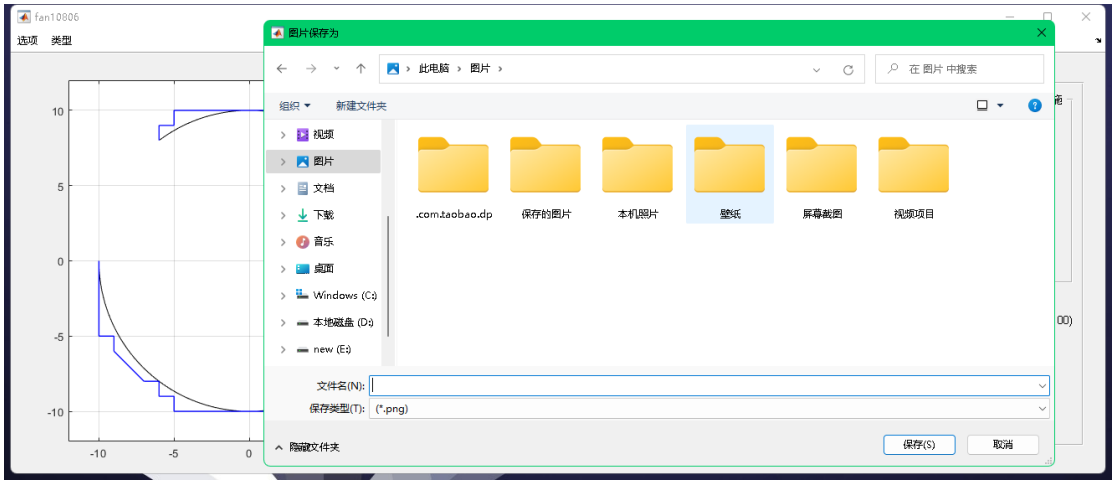


顺时针优弧插补图



逆时针优弧插补图

4.3 插补路径图片保存功能



五、课程设计心得

5.1 课程设计中遇到的问题及解决方法

（1）圆弧全象限插补时情况过多，计算每个象限应该插补的步数时经常打错一个变量导致插补步数计算出错而导致圆弧无法完成插补。我选择关闭插补功能直接输出插补步数的值经过多次反复调试确定没问题了才进行圆弧插补。

（2）圆弧插补权象限容易出现跨坐标轴时跑飞问题，经过比较发现跨象限时实际的 X 与 Y 坐标总满足一个大于（小于）0，另一个大于等于（小于等于）0 的条件，索性整个程序每次插补一个点时都判断一下这个点在第几象限再进行插补工作，最后只要每个象限的插补步数算出来了就会使得插补的程序比较简单。

（3）直线单步插补只能插补一步，后续没有反应。设置全局变量来传递参数。

（4）逐点法插补整圆没反应。设置检测插补对象是否为整圆，若是则开始主动插补一步，后续按照正常逻辑插补成果解决了这个问题。

（5）DDA 圆弧插补时某些半径的圆跨象限时 x,y 先后到达坐标轴导致被积函数寄存器为零累加器无法溢出而导致某一步无法实现而进入死循环的问题。例如在插补 12 象限起点(5,10)终点(-5,10)的逆圆弧时由于 DDA 插补时 x 先到达坐标轴导致 Jvy=0 使得 Jry 始终无法溢出而此时计算的步数 Ey(1)=1，始终无法按照规则减去一而得零，我采用不同跨象限时先检测 x 或 y 是否到达坐标轴来跳过类似 Ey(1)=1 的最后一步使得程序可以继续执行。

5.2 人机交互性

（1）当直线插补输入的数据是错误的时候会发出警告

（2）当步长无法插补时发出警告

（3）插补速度可通过滑条或者输入框两种方式进性调节

（4）输入的圆弧的信息进行自动判定并在信息有误时发出警告

（5）可自动计算圆弧的半径

（6）可以保存插补的路径图像

5.3 总结体会

本次课设我的课设要求比较简单，花了一天熟悉环境做完了直线的逐点法插补和 DDA 法插补，总的来说 matlab 有的功能有点忘了，所以比较慢。然后花了半天就完成了自己的逐点法插补圆弧部分任务，并且实现了全象限插补。不过后来调试 DDA 法圆弧插补时花了大量的时间，感觉 DDA 圆弧插补相较于逐点法难了太多，刚开始我是用的和逐点法一样的判断条件判断 DDA 法插补到达终点，不过后来发现这样做无法正确到达目标点，因为逐点法上的点如果在圆上插补的线段总能到达该点，而数字积分法却不同，本来可能并不可以到达该点（一个坐标的进给完成后需要停止该方向的累加而该判断条件无法方便的判断），所以改为用步数作为终止条件。

在整个过程中，我对插补的概念有了更深的理解，并且由于自己动手写了程序更加直观地了解了插补的全过程，对各种情况如何处理的细节有了更好的把握。课堂上讲解的例子都 是单步长的，并且给出了简化公式，实际编程中由于步长可变所以更需要对插补的原理加深理解。同时在这个过程里我也更好地理解到左移规格化、加载对于插补的意义。

虽然以前用过 MATLAB，但是之前做的机械原理课设并没有单步插补这种功能，这还是第一次接触内心还是挺难的。通过这次课设我不仅复习和运用了上学期选修课所学的 MATLAB 知识，同时还通过网络搜索学习到了更多 MATLAB 有关的命令和功能，进一步探索了强大的 MATLAB 程序。

最后，对帮助我的同学以及指导老师缪群华表达衷心的感谢！

六、参考文献

[1] 陈蔚芳、王宏涛主编.机床数控技术及应用（第四版）.北京：科学出版社，2016

[2] 刘卫国主编.MATLAB 程序设计与应用（第 3 版）.北京：高等教育出版社，

附录一 变量说明

加载模式	loadmode
左移规格化	leftmode
插补模式	ruinmode
直线起点坐标	X0
直线起点坐标	Y0
直线终点坐标	Xe
直线终点坐标	Ye
理论直线	line
圆弧起点坐标	CX0
圆弧起点坐标	CY0
圆弧终点坐标	CXe
圆弧终点坐标	CYe
圆心坐标	CXr
圆心坐标	CYr
圆弧半径	CR
寄存器位数	regbit
插补步长	steplen
圆弧方向	clockwise
偏差判别	F
X 累加器溢出	xflag
Y 累加器溢出	yflag
X 方向插补步数	Ex
Y 方向插补步数	Ey
X 累加器	Jry
Y 累加器	Jrx
X 函数寄存器	Jvx
Y 函数寄存器	Jvy
插补速度	speed

附录二 主要源程序

理论直线绘制

```
global X0 Y0 Xe Ye;
axes(handles.axes1);
if (isnan(X0)||isnan(Y0)||isnan(Xe)||isnan(Ye))
    warndlg('请输入数字','警告','on');
    return;
end
if ((X0==Xe)&&(Y0==Ye))
     warndlg('首尾坐标相同，输入无效，请重新输入','警告','on');
     return;
end

plot([X0,Xe],[Y0,Ye],'k');
dx=abs(X0-Xe);
dy=abs(Y0-Ye);
if dx>dy
   axis([min([X0,Xe])-dx*0.2,min([X0,Xe])+dx*1.2,min([Y0,Ye])-
dx*0.2,min([Y0,Ye])+dx*1.2]);
else
   axis([min([X0,Xe])-dy*0.2,min([X0,Xe])+dy*1.2,min([Y0,Ye])-
dy*0.2,min([Y0,Ye])+dy*1.2]);
end

axis equal;
grid on;
grid minor;
hold on;
```

理论圆弧绘制

```
global CX0 CY0 CXe CYe CR CXr CYr CA0 CAe clockwise;
axes(handles.axes1);
if
(isnan(CX0)||isnan(CY0)||isnan(CXe)||isnan(CYe))||isnan(CXr)||isnan(C
Yr)||isnan(clockwise)
    warndlg('请输入数字','警告','on');
    return;
end
if ((CX0-CXr)^2+(CY0-CYr)^2)~=((CXe-CXr)^2+(CYe-CYr)^2)
    warndlg('圆心坐标错误','警告','on');
```

```
    return;
else
    CR=sqrt((CXO-CXr)^2+(CYO-CYr)^2);
    set(handles.CR,'string',num2str(CR));
    CXOr=CXO-CXr;
    CYOr=CYO-CYr;
    CXer=CXe-CXr;
    CYer=CYe-CYr;
    if CXOr>0
        if CYOr>0
            CAO=abs(atan(CYOr./CXOr));
        elseif CYOr==0
            CAO=0;
        else
            CAO=2*pi-abs(atan(CYOr./CXOr));
        end
    elseif CXOr==0
        if CYOr>0
            CAO=pi/2;
        else
            CAO=pi*3/2;
        end
    else
        if CYOr>0
            CAO=pi-abs(atan(CYOr./CXOr));
        elseif CYOr==0
            CAO=pi;
        else
            CAO=pi+abs(atan(CYOr./CXOr));
        end
    end
    if CXer>0
        if CYer>0
            CAe=abs(atan(CYer./CXer));
        elseif CYer==0
            CAe=0;
        else
            CAe=2*pi-abs(atan(CYer./CXer));
        end
    elseif CXer==0
        if CYer>0
            CAe=pi/2;
        else
            CAe=pi*3/2;
```

```
        end
    else
        if CYer>0
            CAe=pi-abs(atan(CYer./CXer));
        elseif CYer==0
            CAe=pi;
        else
            CAe=pi+abs(atan(CYer./CXer));
        end
    end

    %理论圆弧绘制
    if clockwise==0  %逆圆弧
        if CAe>CA0
            t=CA0:0.01:CAe;
        else
            t=CA0:0.01:CAe+2*pi;
        end
    else
        if CAe>=CA0
            t=CAe:0.01:CA0+2*pi;
        else
            t=CAe:0.01:CA0;
        end
    end
    cx=CR*cos(t)+CXr;
    cy=CR*sin(t)+CYr;
    plot(cx,cy,'black');%,'linewidth',1.5
    hold on;
    plot(CXr,CYr)
    axis([CXr-CR*1.2,CXr+CR*1.2,CYr-CR*1.2,CYr+CR*1.2]);
    axis square;
    grid on;
    hold on;
end
```

直线插补

```
global X0 Y0 Xe Ye regbit steplen speed ruinmode loadmode leftmode;
axes(handles.axes1);
PX1=X0;
PY1=Y0;
PX2=X0;
PY2=Y0;
```

```
Jvx=abs(X0-Xe);
Jvy=abs(Y0-Ye);
Jrx=0;
Jry=0;
Ex0=abs(PX1-Xe);
Ey0=abs(PY1-Ye);
OF=2^regbit;
if (Xe>X0&&Ye>Y0)   %第一象限
   t=1;
end
if (Xe<X0&&Ye>Y0)   %第二象限
   t=2;
end
if (Xe<X0&&Ye<Y0)   %第三象限
   t=3;
end
if (Xe>X0&&Ye<Y0)   %第四象限
   t=4;
end
if  fix(abs(X0-Xe)/steplen)~=abs(X0-Xe)/steplen    ||    fix(abs(Y0-
Ye)/steplen)~=abs(Y0-Ye)/steplen
    warndlg("进给步长错误无法完成差补")
    return;
end

if ruinmode==0 %DDA法
    if Jvx>OF || Jvy>OF
        warndlg("寄存器位数错误无法完成差补")
        return;
    end
    if loadmode==1
        Jrx=OF/2;
        Jry=OF/2;
    end
    if loadmode==2
        Jrx=OF-1;
        Jry=OF-1;
    end
    if leftmode==1
        while Jvx<OF/2&&Jvy<OF/2
        Jvx=Jvx*2;
        Jvy=Jvy*2;
        end
    end
```

```
switch t
case 1 %第一象限
    while Ex0>0.5*steplen||Ey0>0.5*steplen
        Jrx=Jrx+Jvx;
        Jry=Jry+Jvy;
        pause(1/speed);
        if Jrx>=OF
            Jrx=Jrx-OF;
            PX2=PX1+steplen;%向+X 进给一个步长
            Ex0=abs(PX2-Xe);
        end
        if  Jry>=OF
            Jry=Jry-OF;
            PY2=PY1+steplen;%向+Y 进给一个步长
            Ey0=abs(PY2-Ye);
        end
        if  (PX2==PX1&&PY2==PY1)
        else
            plot([PX1,PX2],[PY1,PY2],'b','linewidth',1);

        end
        PX1=PX2;
        PY1=PY2;
    end
case 2 %第二象限
    while Ex0>0.5*steplen||Ey0>0.5*steplen
        Jrx=Jrx+Jvx;
        Jry=Jry+Jvy;
        pause(1/speed);
        if Jrx>=OF
            Jrx=Jrx-OF;
            PX2=PX1-steplen;%向-X 进给一个步长
            Ex0=abs(PX2-Xe);
        end
        if  Jry>=OF
            Jry=Jry-OF;
            PY2=PY1+steplen;%向+Y 进给一个步长
            Ey0=abs(PY2-Ye);
        end
        if  (PX2==PX1&&PY2==PY1)
        else
            plot([PX1,PX2],[PY1,PY2],'b','linewidth',1);

        end
```

```
        PX1=PX2;
        PY1=PY2;
    end
case 3 %第三象限
    while Ex0>0.5*steplen||Ey0>0.5*steplen
        Jrx=Jrx+Jvx;
        Jry=Jry+Jvy;
        pause(1/speed);
        if Jrx>=OF
            Jrx=Jrx-OF;
            PX2=PX1-steplen;%向-X 进给一个步长
            Ex0=abs(PX2-Xe);
        end
        if  Jry>=OF
            Jry=Jry-OF;
            PY2=PY1-steplen;%向-Y 进给一个步长
            Ey0=abs(PY2-Ye);
        end
        if  (PX2==PX1&&PY2==PY1)
        else
            plot([PX1,PX2],[PY1,PY2],'b','linewidth',1);

        end
        PX1=PX2;
        PY1=PY2;
    end
case 4 %第四象限
    while Ex0>0.5*steplen||Ey0>0.5*steplen
        Jrx=Jrx+Jvx;
        Jry=Jry+Jvy;
        pause(1/speed);
        if Jrx>=OF
            Jrx=Jrx-OF;
            PX2=PX1+steplen;   %向+X 进给一个步长
            Ex0=abs(PX2-Xe);
        end
        if  Jry>=OF
            Jry=Jry-OF;
            PY2=PY1-steplen;   %向-Y 进给一个步长
            Ey0=abs(PY2-Ye);
        end
        if  (PX2==PX1&&PY2==PY1)
        else
            plot([PX1,PX2],[PY1,PY2],'b','linewidth',1);
```

```
            end
            PX1=PX2;
            PY1=PY2;
        end
    end
end
if ruinmode==1 %逐点法
    X=X0;
    Y=Y0;
    F=0;
    Ex1=abs(X-Xe);
    Ey1=abs(Y-Ye);
    switch t
        case 1  %第一象限
            while Ex1>0.5*steplen || Ey1>0.5*steplen
                if F>=0
                    plot([X,X+steplen],[Y,Y],'r','linewidth',1);
                    pause(1/speed);
                    F=F-abs(Ye-Y0)*steplen;
                    X=X+steplen;
                    Ex1=abs(X-Xe);
                else
                    plot([X,X],[Y,Y+steplen],'r','linewidth',1);
                    pause(1/speed);
                    F=F+abs(Xe-X0)*steplen;
                    Y=Y+steplen;
                    Ey1=abs(Y-Ye);
                end
            end
        case 2  %第二象限
            while Ex1>0.5*steplen||Ey1>0.5*steplen
                if F>=0
                    plot([X,X-steplen],[Y,Y],'r','linewidth',1);
                    pause(1/speed);
                    F=F-abs(Ye-Y0)*steplen;
                    X=X-steplen;
                    Ex1=abs(X-Xe);
                else
                    plot([X,X],[Y,Y+steplen],'r','linewidth',1');
                    pause(1/speed);
                    F=F+abs(Xe-X0)*steplen;
                    Y=Y+steplen;
                    Ey1=abs(Y-Ye);
                end
```

```
            end
        case 3  %第三象限
            while Ex1>0.5*steplen||Ey1>0.5*steplen
                if F>=0
                    plot([X,X-steplen],[Y,Y],'r','linewidth',1);
                    pause(1/speed);
                    F=F-abs(Ye-Y0)*steplen;
                    X=X-steplen;
                    Ex1=abs(X-Xe);
                else
                    plot([X,X],[Y,Y-steplen],'r','linewidth',1);
                    pause(1/speed);
                    F=F+abs(Xe-X0)*steplen;
                    Y=Y-steplen;
                    Ey1=abs(Y-Ye);
                end
            end
         case 4  %第四象限
            while Ex1>0.5*steplen||Ey1>0.5*steplen
                if F>=0
                    plot([X,X+steplen],[Y,Y],'r','linewidth',1);
                    pause(1/speed);
                    F=F-abs(Ye-Y0)*steplen;
                    X=X+steplen;
                    Ex1=abs(X-Xe);
                else
                    plot([X,X],[Y,Y-steplen],'r','linewidth',1);
                    pause(1/speed);
                    F=F+abs(Xe-X0)*steplen;
                    Y=Y-steplen;
                    Ey1=abs(Y-Ye);
                end
            end
    end
end
```

圆弧插补

```
global CX0 CY0 CXr CYr CXe CYe steplen clockwise speed ruinmode
regbit CR loadmode leftmode;
X=CX0;              %逐点法当前 X 坐标
Y=CY0;              %逐点法当前 Y 坐标
CPx=X-CXr;          %逐点法当前相对 X 坐标
```

```
CPy=Y-CYr;          %逐点法当前相对 X 坐标
dx=steplen;
dy=steplen;
F=0;                %逐点法判别式 F
cnt = 0;            %插补整圆
if fix(abs(CXO-CXe)/steplen) ~= abs(CXO-CXe)/steplen || fix(abs(CYO-
CYe)/steplen) ~= abs(CYO-CYe)/steplen
    warndlg("进给步长错误无法完成差补")
    return;
end
if ruinmode==1 %逐点法
    if clockwise==0   %逆圆弧
        while (sqrt((X-CXe)^2+(Y-CYe)^2))>=steplen || cnt==0
            cnt = 1;
            if CPx>0 && CPy>=0   %1 象限
                if F>=0
                    F=F-2*steplen*CPx+steplen^2;
                    plot([X,X-dx],[Y,Y],'b','linewidth',1)
                    X=X-dx;
                    pause(1/speed);
                else
                    F=F+2*steplen*CPy+steplen^2;
                    plot([X,X],[Y,Y+dy],'b','linewidth',1)
                    Y=Y+dy;
                    pause(1/speed);
                end
            end
            if CPx<=0 && CPy>0   %2 象限
                if F>=0
                    F=F-2*steplen*CPy+steplen^2;
                    plot([X,X],[Y,Y-dy],'b','linewidth',1)
                    Y=Y-dy;
                    pause(1/speed);
                else
                    F=F-2*steplen*CPx+steplen^2;
                    plot([X,X-dx],[Y,Y],'b','linewidth',1)
                    X=X-dx;
                    pause(1/speed);
                end
            end
            if CPx<0 && CPy<=0   %3 象限
                if F>=0
                    F=F+2*steplen*CPx+steplen^2;
                    plot([X,X+dx],[Y,Y],'b','linewidth',1)
```

```
                X=X+dx;
                pause(1/speed);
            else
                F=F-2*CPy*steplen+steplen^2;
                plot([X,X],[Y,Y-dy],'b','linewidth',1)
                Y=Y-dy;
                pause(1/speed);
            end
        end
        if CPx>=0 && CPy<0                    %4 象限
            if F>=0
                F=F+2*steplen*CPy+steplen^2;
                plot([X,X],[Y,Y+dy],'b','linewidth',1)
                Y=Y+dy;
                pause(1/speed);
            else
                F=F+2*steplen*CPx+steplen^2;
                plot([X,X+dx],[Y,Y],'b','linewidth',1)
                X=X+dx;
                pause(1/speed);
            end
        end
        CPx=X-CXr;
        CPy=Y-CYr;
    end
else
    while (sqrt((X-CXe)^2+(Y-CYe)^2))>=steplen || cnt == 0
        cnt = 1;
        if CPx>0 && CPy<=0   %4 象限
            if F>=0
                F=F-2*steplen*CPx+steplen^2;
                plot([X,X-dx],[Y,Y],'b','linewidth',1)
                X=X-dx;
                pause(1/speed);
            else
                F=F-2*steplen*CPy+steplen^2;
                plot([X,X],[Y,Y-dy],'b','linewidth',1)
                Y=Y-dy;
                pause(1/speed);
            end
        end
        if CPx<=0 && CPy<0   %3 象限
            if F>=0
                F=F+2*steplen*CPy+steplen^2;
```

```
                    plot([X,X],[Y,Y+dy],'b','linewidth',1)
                    Y=Y+dy;
                    pause(1/speed);
                else
                    F=F-2*steplen*CPx+steplen^2;
                    plot([X,X-dx],[Y,Y],'b','linewidth',1)
                    X=X-dx;
                    pause(1/speed);
                end
            end
            if CPx<0 && CPy>=0   %2 象限
                if F>=0
                    F=F+2*steplen*CPx+steplen^2;
                    plot([X,X+dx],[Y,Y],'b','linewidth',1)
                    X=X+dx;
                    pause(1/speed);
                else
                    F=F+2*steplen*CPy+steplen^2;
                    plot([X,X],[Y,Y+dy],'b','linewidth',1)
                    Y=Y+dy;
                    pause(1/speed);
                end
            end
            if CPx>=0 && CPy>0   %1 象限
                if F>=0
                    F=F-2*steplen*CPy+steplen^2;
                    plot([X,X],[Y,Y-dy],'b','linewidth',1)
                    Y=Y-dy;
                    pause(1/speed);
                else
                    F=F+2*steplen*CPx+steplen^2;
                    plot([X,X+dx],[Y,Y],'b','linewidth',1)
                    X=X+dx;
                    pause(1/speed);
                end
            end
            CPx=X-CXr;
            CPy=Y-CYr;
        end
    end
end
if ruinmode==0 %DDA 法
    PX1=CX0;
    PY1=CY0;
```

```
    PX2=CX0;
    PY2=CY0;
    CPx=CX0-CXr;
    CPy=CY0-CYr;
    CPX0=CX0-CXr;
    CPY0=CY0-CYr;
    CPXe=CXe-CXr;
    CPYe=CYe-CYr;
    dx=steplen;
    dy=steplen;
    xflag=1;    %Jrx 累加开关
    yflag=1;    %Jry 累加开关
    Jvx=abs(CPY0);   %y 相对值
    Jvy=abs(CPX0);   %x 相对值
    Jrx=0;
    Jry=0;
    OF=2^regbit;
    Cinf=0;
    CEx = [0 0 0 0];
    CEy = [0 0 0 0];
    if CR/steplen>OF
        warndlg("寄存器位数错误无法完成差补")
        return;
    end
    if   fix(abs(CX0-CXr)/steplen)   ~=   abs(CX0-CXr)/steplen   ||
fix(abs(CY0-CYr)/steplen) ~= abs(CY0-CYr)/steplen
        warndlg("DDA 进给步长错误无法完成差补")
        return;
    end
    if loadmode==1
        Jrx=OF/2;
        Jry=OF/2;
    end
    if loadmode==2
        Jrx=OF-1;
        Jry=OF-1;
    end
    if leftmode==1
        while Jvx<OF/4&&Jvy<OF/4
            Jvx=Jvx*2;
            Jvy=Jvy*2;
            dx=dx*2;
            dy=dy*2;
        end
```

```
end
if CXO-CXe==0 && CYO==CYe
    warndlg("不支持整圆插补")
    return;
end
%插补准备判断象限计算步数
if clockwise==0   %逆圆弧
    if CPXO>0 && CPYO>=0        %逆圆弧 1 象限
        Cinf=110;
    elseif CPXO<=0 && CPYO>0    %逆圆弧 2 象限
        Cinf=120;
    elseif CPXO<0 && CPYO<=0    %逆圆弧 3 象限
        Cinf=130;
    elseif CPXO>=0 && CPYO<0    %逆圆弧 4 象限
        Cinf=140;
    end
else              %顺圆弧
    if CPXO>0 && CPYO<=0        %顺圆弧 4 象限
        Cinf=240;
    elseif CPXO<=0 && CPYO<0    %顺圆弧 3 象限
        Cinf=230;
    elseif CPXO<0 && CPYO>=0    %顺圆弧 2 象限
        Cinf=220;
    elseif CPXO>=0 && CPYO>0    %顺圆弧 1 象限
        Cinf=210;
    end
end

%终点坐标判别
if clockwise==0   %逆圆弧
    if CPXe>=0 && CPYe>0        %1 象限
        Cinf=Cinf+1;
    elseif CPXe<0 && CPYe>=0    %2 象限
        Cinf=Cinf+2;
    elseif CPXe<=0 && CPYe<0    %3 象限
        Cinf=Cinf+3;
    elseif CPXe>0 && CPYe<=0    %4 象限
        Cinf=Cinf+4;
    end
else
    if CPXe>0 && CPYe>=0        %1 象限
        Cinf=Cinf+1;
    elseif CPXe<=0 && CPYe>0    %2 象限
        Cinf=Cinf+2;
```

```
        elseif CPXe<0 && CPYe<=0   %3 象限
            Cinf=Cinf+3;
        elseif CPXe>=0 && CPYe<0   %4 象限
            Cinf=Cinf+4;
        end
    end

    switch Cinf
        case 111    %逆圆弧 1-->1
            if CXO-CXe<0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=ceil(abs(CXO-
CXe)/steplen);CEx(2)=0;CEx(3)=0;CEx(4)=0;
            CEy(1)=ceil(abs(CYO-
CYe)/steplen);CEy(2)=0;CEy(3)=0;CEy(4)=0;
        case 122
            if CXO-CXe<0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=0;CEx(2)=ceil(abs(CXO-
CXe)/steplen);CEx(3)=0;CEx(4)=0;
            CEy(1)=0;CEy(2)=ceil(abs(CYO-
CYe)/steplen);CEy(3)=0;CEy(4)=0;
        case 133
            if CXO-CXe>0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=0;CEx(2)=0;CEx(3)=ceil(abs(CXO-
CXe)/steplen);CEx(4)=0;
            CEy(1)=0;CEy(2)=0;CEy(3)=ceil(abs(CYO-
CYe)/steplen);CEy(4)=0;
        case 144
            if CXO-CXe>0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=0;CEx(2)=0;CEx(3)=0;CEx(4)=ceil(abs(CXO-
CXe)/steplen);
            CEy(1)=0;CEy(2)=0;CEy(3)=0;CEy(4)=ceil(abs(CYO-
CYe)/steplen);
```

```
        case 112

CEx(1)=ceil(abs(CPX0)/steplen);CEx(2)=ceil(abs(CPXe)/steplen);CEx(3)=
0;CEx(4)=0;
            CEy(1)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(2)=ceil(abs(abs(CPYe)-CR)/steplen);CEy(3)=0;CEy(4)=0;
        case 123
            CEx(1)=0;CEx(2)=ceil(abs(abs(CPX0)-
CR)/steplen);CEx(3)=ceil(abs(abs(CPXe)-CR)/steplen);CEx(4)=0;

CEy(1)=0;CEy(2)=ceil(abs(CPY0)/steplen);CEy(3)=ceil(abs(CPYe)/steplen
);CEy(4)=0;
        case 134

CEx(1)=0;CEx(2)=0;CEx(3)=ceil(abs(CPX0)/steplen);CEx(4)=ceil(abs(CPXe
)/steplen);
            CEy(1)=0;CEy(2)=0;CEy(3)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(4)=ceil(abs(abs(CPYe)-CR)/steplen);
        case 141
            CEx(1)=ceil(abs(abs(CPXe)-
CR)/steplen);CEx(2)=0;CEx(3)=0;CEx(4)=ceil(abs(abs(CPX0)-CR)/steplen);

CEy(1)=ceil(abs(CPYe)/steplen);CEy(2)=0;CEy(3)=0;CEy(4)=ceil(abs(CPY0
)/steplen);
        case 113

CEx(1)=ceil(abs(CPX0)/steplen);CEx(2)=ceil(CR/steplen);CEx(3)=ceil(ab
s(abs(CPXe)-CR)/steplen);CEx(4)=0;
            CEy(1)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(2)=ceil(CR/steplen);CEy(3)=ceil(abs(CPYe)/steplen);C
Ey(4)=0;
        case 124
            CEx(1)=0;CEx(2)=ceil(abs(abs(CPX0)-
CR)/steplen);CEx(3)=ceil(CR/steplen);CEx(4)=ceil(abs(CPXe)/steplen);

CEy(1)=0;CEy(2)=ceil(abs(CPY0)/steplen);CEy(3)=ceil(CR/steplen);CEy(4
)=ceil(abs(abs(CPYe)-CR)/steplen);
        case 131
            CEx(1)=ceil(abs(abs(CPXe)-
CR)/steplen);CEx(2)=0;CEx(3)=ceil(abs(CPX0)/steplen);CEx(4)=ceil(CR/s
teplen);

CEy(1)=ceil(abs(CPYe)/steplen);CEy(2)=0;CEy(3)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(4)=ceil(CR/steplen);
```

```
        case 142

CEx(1)=ceil(CR/steplen);CEx(2)=ceil(abs(CPXe)/steplen);CEx(3)=0;CEx(4
)=ceil(abs(abs(CPX0)-CR)/steplen);
            CEy(1)=ceil(CR/steplen);CEy(2)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(3)=0;CEy(4)=ceil(abs(CPY0)/steplen);
        case 114

CEx(1)=ceil(abs(CPX0)/steplen);CEx(2)=ceil(CR/steplen);CEx(3)=ceil(CR
/steplen);CEx(4)=ceil(abs(CPXe)/steplen);
            CEy(1)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(2)=ceil(CR/steplen);CEy(3)=ceil(CR/steplen);CEy(4)=c
eil(abs(abs(CPYe)-CR)/steplen);
        case 121
            CEx(1)=ceil(abs(abs(CPXe)-
CR)/steplen);CEx(2)=ceil(abs(abs(CPX0)-
CR)/steplen);CEx(3)=ceil(CR/steplen);CEx(4)=ceil(CR/steplen);

CEy(1)=ceil(abs(CPYe)/steplen);CEy(2)=ceil(abs(CPY0)/steplen);CEy(3)=
ceil(CR/steplen);CEy(4)=ceil(CR/steplen);
        case 132

CEx(1)=ceil(CR/steplen);CEx(2)=ceil(abs(CPXe)/steplen);CEx(3)=ceil(ab
s(CPX0)/steplen);CEx(4)=ceil(CR/steplen);
            CEy(1)=ceil(CR/steplen);CEy(2)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(3)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(4)=ceil(CR/steplen);
        case 143

CEx(1)=ceil(CR/steplen);CEx(2)=ceil(CR/steplen);CEx(3)=ceil(abs(abs(C
PXe)-CR)/steplen);CEx(4)=ceil(abs(abs(CPX0)-CR)/steplen);

CEy(1)=ceil(CR/steplen);CEy(2)=ceil(CR/steplen);CEy(3)=ceil(abs(CPYe)
/steplen);CEy(4)=ceil(abs(CPY0)/steplen);
        %----------------------顺圆弧--------------------------------
        case 211    %顺圆弧 1-->1
            if CX0-CXe>0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=ceil(abs(CX0-
CXe)/steplen);CEx(2)=0;CEx(3)=0;CEx(4)=0;
            CEy(1)=ceil(abs(CY0-
CYe)/steplen);CEy(2)=0;CEy(3)=0;CEy(4)=0;
```

```
        case 222
            if CX0-CXe>0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=0;CEx(2)=ceil(abs(CX0-
CXe)/steplen);CEx(3)=0;CEx(4)=0;
            CEy(1)=0;CEy(2)=ceil(abs(CY0-
CYe)/steplen);CEy(3)=0;CEy(4)=0;
        case 233
            if CX0-CXe<0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=0;CEx(2)=0;CEx(3)=ceil(abs(CX0-
CXe)/steplen);CEx(4)=0;
            CEy(1)=0;CEy(2)=0;CEy(3)=ceil(abs(CY0-
CYe)/steplen);CEy(4)=0;
        case 244
            if CX0-CXe<0
                warndlg("不支持该插补类型")
                return;
            end
            CEx(1)=0;CEx(2)=0;CEx(3)=0;CEx(4)=ceil(abs(CX0-
CXe)/steplen);
            CEy(1)=0;CEy(2)=0;CEy(3)=0;CEy(4)=ceil(abs(CY0-
CYe)/steplen);
        case 214        %1-4
            CEx(1)=ceil(abs(abs(CPX0)-
CR)/steplen);CEx(2)=0;CEx(3)=0;CEx(4)=ceil(abs(abs(CPXe)-CR)/steplen);

CEy(1)=ceil(abs(CPY0)/steplen);CEy(2)=0;CEy(3)=0;CEy(4)=ceil(abs(CPYe
)/steplen);
        case 243

CEx(1)=0;CEx(2)=0;CEx(3)=ceil(abs(CPXe)/steplen);CEx(4)=ceil(abs(CPX0
)/steplen);
            CEy(1)=0;CEy(2)=0;CEy(3)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(4)=ceil(abs(abs(CPY0)-CR)/steplen);
        case 232
            CEx(1)=0;CEx(2)=ceil(abs(abs(CPXe)-
CR)/steplen);CEx(3)=ceil(abs(abs(CPX0)-CR)/steplen);CEx(4)=0;

CEy(1)=0;CEy(2)=ceil(abs(CPYe)/steplen);CEy(3)=ceil(abs(CPY0)/steplen
```

```
);CEy(4)=0;
        case 221

CEx(1)=ceil(abs(CPXe)/steplen);CEx(2)=ceil(abs(CPX0)/steplen);CEx(3)=
0;CEx(4)=0;
            CEy(1)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(2)=ceil(abs(abs(CPY0)-CR)/steplen);CEy(3)=0;CEy(4)=0;
        case 213        %1-4-3
            CEx(1)=ceil(abs(abs(CPX0)-
CR)/steplen);CEx(2)=0;CEx(3)=ceil(abs(CPXe)/steplen);CEx(4)=ceil(CR/s
teplen);

CEy(1)=ceil(abs(CPY0)/steplen);CEy(2)=0;CEy(3)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(4)=ceil(CR/steplen);
        case 224        %2-1-4

CEx(1)=ceil(CR/steplen);CEx(2)=ceil(abs(CPX0)/steplen);CEx(3)=0;CEx(4
)=ceil(abs(abs(CPXe)-CR)/steplen);
            CEy(1)=ceil(CR/steplen);CEy(2)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(3)=0;CEy(4)=ceil(abs(CPYe)/steplen);
        case 231        %3-2-1

CEx(1)=ceil(abs(CPXe)/steplen);CEx(2)=ceil(CR/steplen);CEx(3)=ceil(ab
s(abs(CPX0)-CR)/steplen);CEx(4)=0;
            CEy(1)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(2)=ceil(CR/steplen);CEy(3)=ceil(abs(CPY0)/steplen);C
Ey(4)=0;
        case 242        %4-3-2
            CEx(1)=0;CEx(2)=ceil(abs(abs(CPXe)-
CR)/steplen);CEx(3)=ceil(CR/steplen);CEx(4)=ceil(abs(CPX0)/steplen);

CEy(1)=0;CEy(2)=ceil(abs(CPYe)/steplen);CEy(3)=ceil(CR/steplen);CEy(4
)=ceil(abs(abs(CPY0)-CR)/steplen);
        case 212        %1-4-3-2
            CEx(1)=ceil(abs(abs(CPX0)-
CR)/steplen);CEx(2)=ceil(abs(abs(CPXe)-
CR)/steplen);CEx(3)=ceil(CR/steplen);CEx(4)=ceil(CR/steplen);

CEy(1)=ceil(abs(CPY0)/steplen);CEy(2)=ceil(abs(CPYe)/steplen);CEy(3)=
ceil(CR/steplen);CEy(4)=ceil(CR/steplen);
        case 223        %2-1-4-3

CEx(1)=ceil(CR/steplen);CEx(2)=ceil(abs(CPX0)/steplen);CEx(3)=ceil(ab
s(CPXe)/steplen);CEx(4)=ceil(CR/steplen);
```

```
            CEy(1)=ceil(CR/steplen);CEy(2)=ceil(abs(abs(CPY0)-
CR)/steplen);CEy(3)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(4)=ceil(CR/steplen);
        case 234        %3-2-1-4

CEx(1)=ceil(CR/steplen);CEx(2)=ceil(CR/steplen);CEx(3)=ceil(abs(abs(C
PX0)-CR)/steplen);CEx(4)=ceil(abs(abs(CPXe)-CR)/steplen);

CEy(1)=ceil(CR/steplen);CEy(2)=ceil(CR/steplen);CEy(3)=ceil(abs(CPY0)
/steplen);CEy(4)=ceil(abs(CPYe)/steplen);
        case 241        %4-3-2-1

CEx(1)=ceil(abs(CPXe)/steplen);CEx(2)=ceil(CR/steplen);CEx(3)=ceil(CR
/steplen);CEx(4)=ceil(abs(CPX0)/steplen);
            CEy(1)=ceil(abs(abs(CPYe)-
CR)/steplen);CEy(2)=ceil(CR/steplen);CEy(3)=ceil(CR/steplen);CEy(4)=c
eil(abs(abs(CPY0)-CR)/steplen);
        otherwise

    end
    fprintf("Cinf: %d\r\n", Cinf);
    disp("CEx");
    disp(CEx);
    disp("CEy");
    disp(CEy);
    if clockwise==0   %逆圆弧
        while  CEx(1)>0  ||  CEx(2)>0  ||  CEx(3)>0  ||  CEx(4)>0  ||
CEy(1)>0 || CEy(2)>0 || CEy(3)>0 || CEy(4)>0
            if xflag
                Jrx=Jrx+Jvx;
            end
            if yflag
                Jry=Jry+Jvy;
            end
            if CPx>0 && CPy>=0   %1 象限
                if Jrx>=OF     %x 溢出
                    Jrx=Jrx-OF;
                    PX2=PX1-steplen;
                    Jvy=Jvy-dx;
                    if CEx(1)>1
                        CEx(1)=CEx(1)-1;
                    else
                        CEx(1)=CEx(1)-1;
                        xflag=0;
```

```
        end
    end
    if Jry>=OF    %y 溢出
        Jry=Jry-OF;
        PY2=PY1+steplen;
        Jvx=Jvx+dy;
        if CEy(1)>1
            CEy(1)=CEy(1)-1;
        else
            CEy(1)=CEy(1)-1;
            yflag=0;
        end
    end
    if PX2-CXr==0 && CEy(1)>0
        CEy(2)=CEy(2)-CEy(1);
        CEy(1)=0;
    end
    if CEx(1)<=0 && CEy(1)<=0
        xflag=1;
        yflag=1;
        Jrx=0;
        Jry=0;
    end
elseif CPx<=0 && CPy>0 %2 象限
    if Jrx>=OF     %x 溢出
        Jrx=Jrx-OF;
        PX2=PX1-steplen;
        Jvy=Jvy+dx;
        if CEx(2)>1
            CEx(2)=CEx(2)-1;
        else
            CEx(2)=CEx(2)-1;
            Jrx=0;
            xflag=0;
        end
    end
    if Jry>=OF    %y 溢出
        Jry=Jry-OF;
        PY2=PY1-steplen;
        Jvx=Jvx-dy;
        if CEy(2)>1
            CEy(2)=CEy(2)-1;
        else
            CEy(2)=CEy(2)-1;
```

```
            Jry=0;
            yflag=0;
        end
    end
    if PY2-CYr==0 && CEx(2)>0
        CEx(3)=CEx(3)-CEx(2);
        CEx(2)=0;
    end
    if CEx(2)<=0 && CEy(2)<=0
        xflag=1;
        yflag=1;
        Jrx=0;
        Jry=0;
    end
elseif CPx<0 && CPy<=0 %3 象限
    if Jrx>=OF     %x 溢出
        Jrx=Jrx-OF;
        PX2=PX1+steplen;
        Jvy=Jvy-dx;
        if CEx(3)>1
            CEx(3)=CEx(3)-1;
        else
            CEx(3)=CEx(3)-1;
            xflag=0;
        end
    end
    if Jry>=OF    %y 溢出
        Jry=Jry-OF;
        PY2=PY1-steplen;
        Jvx=Jvx+dy;
        if CEy(3)>1
            CEy(3)=CEy(3)-1;
        else
            CEy(3)=CEy(3)-1;
            yflag=0;
        end
    end

    if PX2-CXr==0 && CEy(3)>0
        CEy(4)=CEy(4)-CEy(3);
        CEy(3)=0;
    end
    if CEx(3)<=0 && CEy(3)<=0
        CEy(3)=0;
```

```
            xflag=1;
            yflag=1;
            Jrx=0;
            Jry=0;
        end

    elseif CPx>=0 && CPy<0 %4 象限
        if Jrx>=OF     %x 溢出
            Jrx=Jrx-OF;
            PX2=PX1+steplen;
            Jvy=Jvy+dx;
            if CEx(4)>1
                CEx(4)=CEx(4)-1;
            else
                CEx(4)=CEx(4)-1;
                xflag=0;
            end
        end
        if Jry>=OF    %y 溢出
            Jry=Jry-OF;
            PY2=PY1+steplen;
            Jvx=Jvx-dy;
            if CEy(4)>1
                CEy(4)=CEy(4)-1;
            else
                CEy(4)=CEy(4)-1;
                yflag=0;
            end
        end

        if PY2-CYr==0 && CEx(4)>0
            CEx(1)=CEx(1)-CEx(4);
            CEx(4)=0;
        end
        if CEx(4)<=0 && CEy(4)<=0
            CEy(4)=0;
            xflag=1;
            yflag=1;
            Jrx=0;
            Jry=0;
        end
    end
    if(PX1 == PX2 && PY1 == PY2)
    else
```

```
                plot([PX1,PX2],[PY1,PY2],'b','linewidth',1);
                pause(1/speed);
            end
            PX1 = PX2;
            PY1 = PY2;
            CPx=PX1-CXr;
            CPy=PY1-CYr;
        end
    else   %顺圆弧
        while  CEx(1)>0  ||  CEx(2)>0  ||  CEx(3)>0  ||  CEx(4)>0  ||
CEy(1)>0 || CEy(2)>0 || CEy(3)>0 || CEy(4)>0
            if xflag
                Jrx=Jrx+Jvx;
            end
            if yflag
                Jry=Jry+Jvy;
            end
            if CPx>=0 && CPy>0   %1 象限
                if Jrx>=OF     %x 溢出
                    Jrx=Jrx-OF;
                    PX2=PX1+steplen;
                    Jvy=Jvy+dx;
                    if CEx(1)>1
                        CEx(1)=CEx(1)-1;
                    else
                        CEx(1)=CEx(1)-1;
                        xflag=0;
                    end
                end
                if Jry>=OF    %y 溢出
                    Jry=Jry-OF;
                    PY2=PY1-steplen;
                    Jvx=Jvx-dy;
                    if CEy(1)>1
                        CEy(1)=CEy(1)-1;
                    else
                        CEy(1)=CEy(1)-1;
                        yflag=0;
                    end
                end

                if PY2-CYr==0 && CEx(1)>0
                    CEx(4)=CEx(4)-CEx(1);
                    CEx(1)=0;
```

```
    end
    if CEx(1)<=0 && CEy(1)<=0
        xflag=1;
        yflag=1;
        Jrx=0;
        Jry=0;
    end
elseif CPx<0 && CPy>=0 %2 象限
    if Jrx>=OF     %x 溢出
        Jrx=Jrx-OF;
        PX2=PX1+steplen;
        Jvy=Jvy-dx;
        if CEx(2)>1
            CEx(2)=CEx(2)-1;
        else
            CEx(2)=CEx(2)-1;
            Jrx=0;
            xflag=0;
        end
    end
    if Jry>=OF    %y 溢出
        Jry=Jry-OF;
        PY2=PY1+steplen;
        Jvx=Jvx+dy;
        if CEy(2)>1
            CEy(2)=CEy(2)-1;
        else
            CEy(2)=CEy(2)-1;
            Jry=0;
            yflag=0;
        end
    end

    if PX2-CXr==0 && CEy(2)>0
        CEy(1)=CEy(1)-CEy(2);
        CEy(2)=0;
    end
    if CEx(2)<=0 && CEy(2)<=0
        xflag=1;
        yflag=1;
        Jrx=0;
        Jry=0;
    end
elseif CPx<=0 && CPy<0 %3 象限
```

```
    if Jrx>=OF      %x 溢出
        Jrx=Jrx-OF;
        PX2=PX1-steplen;
        Jvy=Jvy+dx;
        if CEx(3)>1
            CEx(3)=CEx(3)-1;
        else
            CEx(3)=CEx(3)-1;
            xflag=0;
        end
    end
    if Jry>=OF    %y 溢出
        Jry=Jry-OF;
        PY2=PY1+steplen;
        Jvx=Jvx-dy;
        if CEy(3)>1
            CEy(3)=CEy(3)-1;
        else
            CEy(3)=CEy(3)-1;
            yflag=0;
        end
    end

    if PY2-CYr==0 && CEx(3)>0
        CEx(2)=CEx(2)-CEx(3);
        CEx(3)=0;
    end
    if CEx(3)<=0 && CEy(3)<=0
        CEy(3)=0;
        xflag=1;
        yflag=1;
        Jrx=0;
        Jry=0;
    end
elseif CPx>0 && CPy<=0 %4 象限
    if Jrx>=OF      %x 溢出
        Jrx=Jrx-OF;
        PX2=PX1-steplen;
        Jvy=Jvy-dx;
        if CEx(4)>1
            CEx(4)=CEx(4)-1;
        else
            CEx(4)=CEx(4)-1;
            xflag=0;
```

```
                end
            end
            if Jry>=OF    %y 溢出
                Jry=Jry-OF;
                PY2=PY1-steplen;
                Jvx=Jvx+dy;
                if CEy(4)>1
                    CEy(4)=CEy(4)-1;
                else
                    CEy(4)=CEy(4)-1;
                    yflag=0;
                end
            end
            if PX2-CXr==0 && CEy(4)>0
                CEy(3)=CEy(3)-CEy(4);
                CEy(4)=0;
            end
            if CEx(4)<=0 && CEy(4)<=0
                CEy(4)=0;
                xflag=1;
                yflag=1;
                Jrx=0;
                Jry=0;
            end
        end
        if(PX1 == PX2 && PY1 == PY2)
            else
                plot([PX1,PX2],[PY1,PY2],'b','linewidth',1);
                pause(1/speed);
        end
        PX1 = PX2;
        PY1 = PY2;
        CPx=PX1-CXr;
        CPy=PY1-CYr;
    end
end
end
```