

```
tic;
format long
%由前面的题目以及几何模型可以估计[3,10]遍历所有可能姿态
p2=3:0.5:10;%0.5 为间隔粗糙取点，大概得出各姿态粗糙区间
j=0;
n(1)=0;
%遍历取点，得出粗糙区间
for i=2:length(p2)
    g=@(x)f(x,p2(i));
    n(i)=numofroot(g,-pi,pi,10^(-4));%计算零点数量
    if n(i-1)~=n(i)%若相邻两个 p_2 分点处函数零点个数不一致，则记录
        j=j+1;
        p2_0(1:2,2*j-1)=[p2(i-1);n(i-1)];
        p2_0(1:2,2*j)=[p2(i);n(i)];
    end
end
p2_0%检查所得区间以及对应函数零点
```

```
p2_0 = 2×12
    3.500000000000000    4.000000000000000    4.500000000000000    5.000000000000000 ...
         0    2.000000000000000    2.000000000000000    4.000000000000000
```

```
%利用函数的连续性
%细分利用到二分法，对于每次得到的相邻区间边界进行二分法划分，以缩小相邻区间边界距离，得到较准确区间
for i=1:j
    dp2(i)=p2_0(1,2*i)-p2_0(1,2*i-1);
end%初始化各相邻区间边界距离
while max(dp2)>=5*10^(-3)%各相邻区间边界距离精确到两位小数
    for i=1:j
        pp2=(p2_0(1,2*i)+p2_0(1,2*i-1))/2;%二分法
        g=@(x)f(x,pp2);
        m=numofroot(g,-pi,pi,10^(-4));
        if m==p2_0(2,2*i-1)%若中点所算得零点个数与左侧相等，则左侧相应区间的右边界拓宽至该点
            p2_0(1:2,2*i-1)=[pp2;m];
        else%反之，则右侧相应区间的左边界拓宽至该点
            p2_0(1:2,2*i)=[pp2;m];
        end
        dp2(i)=p2_0(1,2*i)-p2_0(1,2*i-1);%计算各相邻区间边界距离
    end
end
p2_0%输出
```

```
p2_0 = 2×12
    3.707031250000000    3.710937500000000    4.863281250000000    4.867187500000000 ...
         0    2.000000000000000    2.000000000000000    4.000000000000000
```

```
toc;
```

历时 26.105525 秒。

```
function out=f(theta,p2)%将 p2 也设为变量，以便对 p2 进行更换
format long;
L1=3;
L2=3*sqrt(2);
```

```

L3=3;
gamma=pi/4;
p1=5;
p3=3;
x1=5;
x2=0;
y2=6;
A2=L3.*cos(theta)-x1;
B2=L3.*sin(theta);
A3=L2.*(cos(theta).*cos(gamma)-sin(theta).*sin(gamma))-x2;
B3=L2.*(cos(theta).*sin(gamma)+sin(theta).*cos(gamma))-y2;
D=2.*(A2.*B3-B2.*A3);
M1=p2.^2-p1.^2-A2.^2-B2.^2;
M2=p3.^2-p1.^2-A3.^2-B3.^2;
N1=B3.*M1-B2.*M2;
N2=-A3.*M1+A2.*M2;
out=N1.^2+N2.^2-p1.^2*D.^2;
end
function m=numofroot(f,a,b,tol)%计算连续函数在某区间内零点的数量
format long;
x=a:tol:b;%对区间进行细分成点， 间距由 tol 控制
m=0;%初始化零点个数
%由于函数连续，对于本题的特殊情况，不存在固定存在的非变号零点，我们仅寻找变号零点即可
for i=1:(length(x)-1)
    p=f(x(i));
    q=f(x(i+1));
    if p>0&&q<0%若相邻两个分点函数值异号，则说明变号零点存在
        m=m+1;
    elseif p<0&&q>0
        m=m+1;
    elseif p==0%若分点处函数值为 0，则计入零点
        m=m+1;
    end
end
if f(b)==0%最后检查未在循环内的边缘点
    m=m+1;
end
end

```